LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-402112

Infrastructure Plan for ASC Petascale Environments

S. Louis, J. Naegle, R. Tomlinson

March 10, 2008

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Infrastructure Plan for ASC Petascale Environments

Version: 1.0

Date: February 29, 2008

LLNL-TR-402112

Document Block

Item	Details
Document Title	Infrastructure Plan for ASC Petascale Environments
Document Type	Level 2 Milestone Formal Deliverable
Originators/Owners	Steve Louis, LLNL, stlouis@llnl.gov John Naegle, SNL, jhnaegl@snl.gov Bob Tomlinson, LANL, bob@lanl.gov
Last Revision	1.0
Revision Date	02/29/2008
Status	Final Report
Release Date:	03/31/2008

Revision History

Revision Level	Date	Description	Change Summary
0.1	*08/31/2007*	Initial Draft	Initial Internal Draft
0.2	*11/09/2007*	Second Draft	Revised Second Draft
0.3	*12/31/2007*	Third Draft	POC Third Draft w/o TWG report revisions
0.4	*01/31/2008*	Fourth Draft	POC Fourth Draft w/o TWG report revisions
1.0	*02/29/2008*	Final Report	Final Report

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Contents

Preface iii

EXECUTIVE SUMMARY 1

1 SCOPE, VISION, CRITICALITY 1

2 DRIVERS AND REQUIREMENTS 2

3 INFRASTRUCTURE COMPONENTS 3

4 INTEGRATION AND SCHEDULING 5

5 CONCLUSIONS 6

BACKGROUND INFORMATION 9

1 PURPOSE OF THIS DOCUMENT 9

2 ORGANIZATION AND CONTENT 9

3 ASC PLATFORM STRATEGY 10

4 SYSTEM AND INFRASTRUCTURE CONCERNS 11

5 USER AND APPLICATION CONCERNS 13

6 PETASCALE USAGE MODELS 15

7 UNIQUE POSITIONING 17

PROGRAMMING ENVIRONMENTS AND TOOLS 19

1 INTRODUCTION AND BACKGROUND 19

2 DEVELOPMENT/DEPLOYMENT AREAS 19

2.1 Programming Models 19
2.2 Correctness Tools 23
2.3 Performance Analysis Tools 26

3 TIMELINE SUMMARY 29

PETASCALE DATA ANALYSIS 32

1 INTRODUCTION AND BACKGROUND 32

2 DEVELOPMENT/DEPLOYMENT AREAS 34

2.1 Investment in Hardware 34
2.2 Continued Investment in Advanced Analysis Software 38

3 TIMELINE SUMMARY 42

I/O, FILE SYSTEMS, AND STORAGE 43

1 INTRODUCTION AND BACKGROUND 43

2 DEVELOPMENT/DEPLOYMENT AREAS 46

2.1 File Systems and I/O 46
2.2 Archive 53

2.3 Broad File Sharing 54
3 TIMELINE SUMMARY 55

NETWORKS AND INTERCONNECTS 57

1 INTRODUCTION AND BACKGROUND 57

2 DEVELOPMENT/DEPLOYMENT AREAS 57

2.1 WAN Interconnect 57
2.2 Resource Interconnect 61
2.3 Internal Interconnect 64

3 TIMELINE SUMMARY 68

SUMMARY AND CONCLUSIONS 69

1 OVERALL INFRASTRUCTURE STRATEGY 69

2 CROSSCUTTING TECHNICAL CONCERNS 70

3 FY2008–FY2016 PLANNING TIMELINE 71

4 LONG-TERM VIEW AND NEXT STEPS 72

APPENDICES 73

A. Acknowledgments 73

B. Acronyms. 75

C. References 77

D. L2 Milestone Text 78

PREFACE

This plan is a formal deliverable for an ASC CSSE/FOUS FOUS (Computational Systems and Software Environment/Facility Operations and User Support) Level 2 Milestone. The plan identifies, assesses, and specifies development and deployment approaches for critical components in four different technical areas: development environments and tools; petascale data analysis; I/O, file systems and archives; and networks and interconnects. It acknowledges and quantifies potential technical gaps or issues, and, where such gaps exist, defines a prioritized approach to closing them. While the formal milestone deliverable (this document) is planned for completion in March 2008, petascale infrastructure components that it describes will be deployed throughout a decade-long time frame. The plan is applicable to multiple ASC petascale platforms deployed during that period, including Roadrunner, Sequoia Initial Delivery and Sequoia final systems, and other potential petascale platforms as described in the recent *ASC Platform Strategy* document.

This plan will be used as technical input to CSSE/FOUS and senior ASC program managers to better inform yearly detailed program planning. Additionally, it will be used to coordinate goals and objectives in separate parts of the ASC program. The CSSE/FOUS program intends to update the plan regularly to reflect technical progress and newly uncovered technical issues. While updated plans will not, themselves, be Level 2 Milestones deliverables, this regular process is necessary to ensure the plan stays current and relevant. This plan includes the following major sections:

Executive Summary

This part provides an overall summary of infrastructure scope, programmatic vision and criticality, drivers and requirements, a definition of key infrastructure component areas, integration, scheduling, risks, and concluding remarks on key questions, petascale computer *ecosystems,* budget realities, and collaborative approaches.

Background Information

This part provides information about the structure of this document, current and planned platform acquisitions, system infrastructure balance, user and application considerations, petascale usage models, and the unique positioning of ASC efforts.

Technical Working Group Reports

This part provides specific reports on the four technical areas. Each report presents an overview, followed by in-depth discussion of major critical themes and their associated challenges, gaps, concerns, strategies, and timelines.

Summary and Conclusions

This part provides a summary of the major petascale infrastructure issues and strategies, including some that crosscut more than one technical area. A timeline is included to demonstrate how planned infrastructure deployments are related to the ASC platform strategy and schedule.

This plan, written by members of the CSSE and FOUS community, is the result of a yearlong effort that began with an initial planning meeting in Las Vegas in February 2007. Acknowledgments and contributing authors are provided in Appendix A. All of the authors donated significant time to work with the Tri-lab Points of Contact and ASC HQ to produce this document. Their contributions to the document are gratefully acknowledged. This planning document has been reviewed and approved by Tri-lab CSSE/FOUS subprogram management and by CSSE/FOUS HQ program management.

This page intentionally left blank.

EXECUTIVE SUMMARY

1 SCOPE, VISION, CRITICALITY

This *Infrastructure Plan for ASC Petascale Environments* identifies, assesses, and specifies development and deployment approaches for critical infrastructure components in four key CSSE/FOUS (Computational Systems and Software Environment/Facility Operations and User Support) technical areas: (1) development environments and tools; (2) petascale data analysis; (3) I/O, file systems and archives; and (4) networks and interconnects. This Plan identifies and quantifies potential technical gaps or issues, and, where they exist, defines a prioritized approach to closing those gaps. While this planning document represents a specific FY08 Level 2 milestone deliverable, the petascale infrastructure components that it describes will be deployed over the next decade, and this plan is applicable to multiple ASC petascale platforms deployed during that time, including Roadrunner, Sequoia Initial Delivery (ID) and Sequoia final systems. The ASC program requires well-integrated infrastructure components to leverage the investments that will be made in petascale computer systems and maximize their impact across the Tri-lab user community. Some of these CSSE/FOUS infrastructure components will be applicable to both ASC capability and capacity systems. However, there may be situations where the user base or the problem-solving capabilities are so unique (or site specific) that integrating specialized components into a more general purpose simulation environment may require additional trade-off cost decisions.

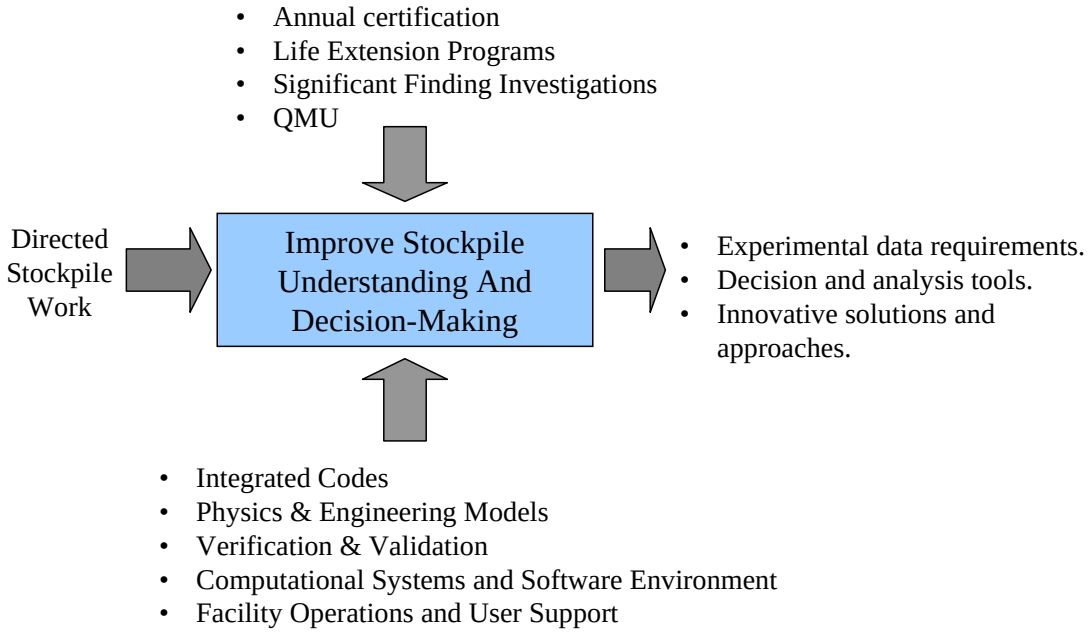
The ASC program's high-performance computational resources are essential enablers for scientists to fulfill stockpile stewardship requirements through the science of predictive simulation in lieu of underground testing. NNSA mission areas that will be positively impacted by the use of increasingly predictive modern simulations on petascale platform acquisitions, and their necessary infrastructure, include support for Significant Finding Investigations (SFIs), weapons certification and annual assessment, Life Extension Programs (LEPs), complex-transforming weapons systems such as the Reliable Replacement Warhead (RRW), and design support for experiments at NNSA experimental facilities. These and other mission areas are more fully described in the *ASC Roadmap: National Nuclear Security through Leadership in Weapons Science* and in the NNSA Predictive Capability Framework (PCF).

The complex and diverse demands that ASC performance and analysis codes will soon place on petascale computational environments and the scale of the required simulations have positioned the ASC program far in advance of the mainstream high-performance computing community. To achieve predictive capability goals, the ASC program must continue to invest in, and influence, the evolution of computational environments, including infrastructure. This requires innovation, tempered by an understanding that computing environments must be stable and must not require applications to be substantially rewritten or reinvented without realizing significant returns. In accordance with NNSA's Complex Transformation vision, ASC will look to operate petascale computing resources as national user facilities, accessible complex-wide, to address the most challenging and pertinent stockpile stewardship issues. Partnering with academia, industry and other federal agencies will likely be needed to develop the required infrastructure to meet the future computing needs of the Nuclear Weapons Complex.

2 DRIVERS AND REQUIREMENTS

Petascale infrastructures developed and deployed through CSSE/FOUS efforts are mandatory for successfully using new ASC petascale platforms in support of improved stockpile understanding. Multi-petaFLOP (PF) platforms are planned for delivery over the next few years, with 100 PF to exascale systems to appear towards the middle and end of the next decade. The principal goal of this document is to define a plan for research, development, deployment, integration and ongoing operation of a well-balanced CSSE/FOUS infrastructure for those new petascale environments. These efforts will not all occur at the same time. Some efforts are specifically targeted shorter-term tactical deployment over the next couple of years. Others are, of necessity, more medium-term strategic research and development with a longer time horizon. Still others will be even longer-term and will need to target architectures that are not yet well defined, but loom on the eventual exascale computing horizon for ASC.

It is critical that we carefully identify CSSE/FOUS infrastructure components most needed by customers and that our efforts be well-managed as an integrated enterprise that meets users' needs. CSSE/FOUS efforts (together with the other major ASC subprograms: Integrated Codes, Physics & Engineering Models, and Verification & Validation) are resources and mechanisms applied to improve stockpile understanding and decisions. ASC products (and the efforts that lead to them) must be structured such that they are clearly explained in a compelling way to stakeholders. In developing new tools and capabilities, output of one activity is vital to the success of others, and it is important to ensure that customer requirements are identified, analyzed, and validated. The diagram below is taken from the *ASC Business Model* (NA-ASC-104R-05-Vol.1-Rev.5), and illustrates this kind of process, with major inputs (left) and outputs (right) depicted as well as requirements (top) and resources/mechanisms (bottom).



Deploying computational environments and user facilities for weapon science studies and other capability computing needs in 2008–2012 is a major goal of the *ASC Roadmap*. The computing resources needed to support ASC in this time frame, as well as in 2013–2015, are

now being identified. In 2005, a study in support of an NNSA DP Level 1 Milestone established that a number of weapon science studies will soon require a petascale computing environment. The demands on these architectures will be used to drive leading edge hardware and software infrastructure technology. Algorithms, tools, and system software will all need to be more fully developed to increase scalability, manage and analyze petascale data sets, and take full advantage of planned petascale platforms.

ASC must continue to engage in partnerships with platform architecture vendors to achieve higher efficiency and improve balance across the range of platform hardware components. ASC will also need to collaborate with industrial, academic, and government partners to build on our "weapons science" computational environments to deliver balanced computing resources over the next decade. This work will also establish the technological foundation to build toward exascale computing environments, which predictive capability will eventually demand.

3 INFRASTRUCTURE COMPONENTS

A summary of major ASC petascale environment infrastructure components and concerns is presented below. More detail can be found in the component-specific technical sections of this document. Note that not all CSSE/FOUS efforts are represented. For example, FOUS efforts for to day-to-day computer center's ongoing system administration, operations and user support at each NNSA laboratory, while critical for overall program success, are not included within the scope of this document.

Programming Models
Programming models and the languages and libraries that implement them must adapt to growing "multi-core" on-chip parallelism and increasing depth of memory hierarchies. Almost all ASC codes currently use similar programming models with MPI for communication. This model has served ASC applications well in cluster and SMP environments over the past decade. However, the enormous parallelism anticipated in new petascale systems is likely to make this "MPI everywhere" model insufficient. The anticipated massive scale of new architectures also raises concerns about power and hardware failure rates. Thus, we must explore and/or develop new petascale programming models.

Correctness Tools
Correctness tools help developers ensure programs run to completion and give the expected result. Needed tools include traditional debuggers, but the scalability of current debugging approaches is unlikely to extend to full petascale systems. Traditional tools also often do not provide adequate insight into root causes of errors. We need to develop and execute an overall correctness tool strategy that combines traditional debuggers with lightweight tools for critical debugging capabilities at large scales while providing more scalability than traditional debuggers and with automated correctness tools that allow greater insight into root causes.

Performance Analysis Tools
Using terascale architectures efficiently has proven challenging, with integrated applications sometimes achieving only a modest percentage of peak performance. Our experience is that this does not necessarily indicate that the system is not being used well. The application may need large numbers of nodes to have sufficient physical memory to run problems of interest,

or the application may be running at the limits of the node's memory bandwidth or latency. Ultimately, the limiting factor on performance is complex and promises to become even more so in the coming petascale era. We need to improve both our current capabilities and our understanding of new petascale performance issues.

Petascale Data Analysis Hardware

Infrastructure investments will be critical for successful petascale data analysis. The Tri-lab strategy of shared resources at the platform site and local resources at individual sites should continue. Shared sites can optimize for a Tri-lab community, while individual sites provide a flexible local environment. This strategy makes effective use of petascale investments, but constraints on I/O can affect the amount of data that can be placed on secondary and tertiary storage, because computational speeds are expected to far outpace our ability to write out and/or visualize data. The speed of data reads from disk will also be a constraint (see below on File Systems and I/O).

Petascale Data Analysis Software

In addition to appropriate investment in hardware to deliver images and data to analysts, we need investment in software tools that can help investigate and understand petascale data. Current distributed memory algorithms and visualization software will scale well on petascale hardware, but it will be increasingly difficult to understand and explore the vast amounts of data in ad-hoc ways. We need to harness the power of computation to assist in analysis of the vast information that petascale computation will produce. This will help leverage ASC investment and will allow customers to spend time thinking rather than wrestling with large data.

File Systems and I/O

To provide effective global parallel file systems, we must address the slow pace at which disk storage devices get faster and more agile, compared to processing advancements. There are multiple I/O patterns in simulation activities that dominate our file system use: N processes writing/reading to 1 shared file; N processes writing/reading to M shared files where M is much less than N; and N processes to N files all into the same directory. There are also write-intensive, parallel checkpoint, defensive I/O workloads that have enormous scale. All of these patterns must now be addressed at petascale.

Archival Systems

Over the past 15 years, HPSS (High-Performance Storage System) has been a dependable high-performance archive service. The archive has grown into the multi-petabyte range and continues to grow on an aggressive curve. Tri-lab investment in HPSS planning, development, and testing keeps HPSS healthy and relevant for our needs. High-performance tape technologies with data rates and densities far better than disk help contribute to the success of the archive service. Unfortunately, the ability to generate massive amounts of information now and at petascale will outstrip infrastructure budget for storage media and must be addressed.

Broad File Sharing

In addition to archives and parallel file systems, there is a need for more general access to file sharing. This service is currently provided by NFSv3. Beyond providing sharing of files on supercomputers, to workstations, and throughout the Tri-labs, there is a desire to reduce the

number of custom file system clients that work with our parallel file systems. The NFS (network file system) protocol family is being targeted for solutions in this space as well. Finally, as potential supercomputer-on-a-chip solutions become prevalent, it is not unreasonable to envision teraflop workstations, which will also need access to our global parallel file systems.

WAN Interconnect

The wide area network (WAN) interconnect between ASC compute facilities consists of leased, private bandwidth and associated equipment to securely interconnect the networks from each of the labs. The bandwidth currently consists of a 10 gigabit/s ring. Because ASC is focused on classified products, WAN links must be protected using approved encryption devices. Beyond encryption, there are other challenges to designing a Tri-lab network that will have large impact on planning and implementation, including geographical distribution of petascale computational resources and the evolving customer usage model for petascale.

Resource Interconnect

The resource interconnect consists of the large network that connects compute platforms to parallel file systems, visualization platforms, pre- and post-processing servers, archival systems, other compute platforms, and the WAN interconnect to other sites. Although each lab has unique implementations of this network, they share key common characteristics: hundreds to a few thousand ports; some form of parallel networking to build the scale required; and Ethernet technology since it scales in distance (fiber) and is common to all platforms. All three labs will face similar challenges and difficulties in addressing future petascale resource interconnects.

Internal Interconnect

For all but the most embarrassingly parallel applications, the ability of the internal high-speed interconnect must be balanced with the rest of the platform architecture to ensure performance at scale. The transition from terascale platforms to petascale is being achieved through higher parallelism within a compute node and by increasing the total number of nodes within a system. Both trends drive the need to increase the performance of traditional high-speed interconnect metrics while creating new requirements that were not significant in the terascale era.

4 INTEGRATION AND SCHEDULING

An objective of this document is to provide collective evidence about how CSSE and FOUS activities and projects support goals of ASC petascale computing over time. This document will also provide ASC executives, program managers, and external audiences with information to help identify and quantify potential technical gaps or issues, and, where they exist, define prioritized approaches and schedules for closing those gaps, potentially through broader industrial, academic and federal agency collaborations. In recent years, efforts were initiated within ASC as platform strategies and their associated computational environments expanded beyond a focus on capability systems to include very large capacity clusters and advanced architectures. However, throughout all of these efforts, there is an understanding of the need for, and benefits of, integrating those platforms (and infrastructures) within a common ASC computational environment. This document attempts to clarify understanding of petascale platform and computational environment development and deployment strategies and point out

how those strategies can become integrated across platforms. It makes more explicit our priorities and motivation, fleshing out the approaches that will form the foundations to achieve the *ASC Roadmap* target of petascale computing in 2009 and to approach a goal of eventual seamless user environments for capability computing in 2013.

This document takes a "phased approach" where research, development, and deployment of infrastructure components are tied to one or more time frames in order to show how and when CSSE/FOUS infrastructures support the ASC program's petascale acquisition and deployment planning.

- Phase I, roughly FY08–FY10, represents a shorter term time frame and includes the early deployment of the LANL Roadrunner system and the LLNL Sequoia ID system.

- Phase II, roughly FY11–FY13, represents a medium term period that includes the LLNL Sequoia Final Delivery platform and possible petascale follow-on capability system(s) to ASC Purple.

- Phase III, roughly FY14–FY16, represents a longer term strategic view for future petascale architectures that are beyond the ASC Programs current platform planning and perhaps beyond current conventional thinking.

The approaches (both tactical and strategic) described in subsequent technical sections of this document addressing infrastructure component technical concerns are mapped to these phases. In addition to phasing and scheduling, the approaches are also prioritized in "importance" and characterized as to levels of "difficulty" and "cost." A particular approach may be applicable to a specific architecture (e.g., a large, homogeneous, many-core platform as envisioned for Sequoia, or a heterogeneous hybrid platform such as Roadrunner), or may be generally applicable to many or all petascale platform possibilities. Some approaches described in this document may be specific to a particular site requirement.

5 CONCLUSIONS

This document presents some answers to questions that were raised in the initial February 2007 meeting to discuss and plan the infrastructure needed for coming ASC petascale environments.

- What are the most important components that must exist for a successful petascale environment, and do any of them overlap technical CSSE/FOUS areas?

- Are there barriers, gaps or issues that must be addressed to develop or deploy these components, and what are the user concerns that motivate these concerns?

- What are the approaches that can be taken to address those barriers, gaps or issues for petascale, and might these approaches be usable and/or relevant outside ASC?

Many answers and strategies will crosscut CSSE/FOUS areas. The following quote is from *Getting Up to Speed: The Future of Supercomputing,* the result of a DOE sponsored National Research Council study published at the beginning of 2005.

> "All aspects of a particular supercomputing ecosystem, be they hardware, software, algorithms or people, must be strong if the ecosystem is to function effectively ... The success of supercomputer architectures is highly dependent on the organisms that form around them."

It has become popular to view supercomputing infrastructure as an ecosystem. Organisms in such an ecosystem are the technologies and components that mutually reinforce each other and the overall stability of the ecosystem. They must adapt and evolve to maintain the health of the ecosystem. Using this analogy, CSSE/FOUS can be viewed as a major piece (or several pieces) of an overall ASC computational ecosystem, together with users, codes, algorithms, and other key "organisms." Surviving and flourishing within such a system requires careful observations, continuous monitoring, and informed decisions regarding choices of technology to maintain an appropriate balance.

Computers today are much faster than before, but harder to use. The ecosystem has moved from sequential codes and vector capability to large parallel clusters (both homogeneous and hybrid) that need MPI communication mechanisms. The number of cores on a die continues to increase; however, processors are only one part of the ecosystem. A balanced petascale ecosystem has requirements that are not necessarily part of lower-end commercial systems and must be driven by (or adapted by) ASC capability and advanced architectures. As described above and in the rest of this document, example areas (or organisms) required by an ASC petascale ecosystem are: programmability and usability of many-core systems through software enhancements and new tools; new processor interconnects that achieve dramatically increased bandwidths and decreased network latencies; reliable and resilient I/O subsystems, file systems and archives; and massive visualization and data analysis capability.

Another conclusion from the National Research Council supercomputer study is that, without large federal investment as a forcing function, the computing industry will not naturally evolve to usable petascale computing systems in the time required for effective and responsible nuclear weapons stewardship. Hence, NNSA will continue to be a major driver for high-end technology. Predictability and continuity of funding will be important prerequisites for success. Unstable funding scenarios or imprudent prioritization of near-term deliverables over more strategic long-term vision may be a detriment to advances in computational capabilities. ASC and NNSA recognize that the nuclear weapons budget in the future is expected to remain constant (or even decline) when adjusted for inflation. NNSA's Complex Transformation vision states that level funding is an overarching constraint for planning purposes, and that should budgets decline sooner than expected, some programs will need to be protected at the expense of schedule, scope, or increased risk. An implication for ASC and CSSE/FOUS is that the Tri-labs are likely to be encouraged to invest in new hardware and software architectural directions in partnership with other federal agencies and computer vendors (whose business plans these investments can leverage) and to develop strategies for productivity gains within realistic future budgets.

This page intentionally left blank.

BACKGROUND INFORMATION

1 PURPOSE OF THIS DOCUMENT

The purpose of the *Infrastructure Plan for ASC Petascale Environments* is to identify, assess, and specify the development and deployment approaches for critical components in four major CSSE technical areas: (1) development environments and tools, (2) petascale data analysis, (3) I/O, file systems, and archives, and (4) networks and interconnects. This plan will identify and quantify potential technical gaps or issues, and, where they exist, will define a prioritized approach to closing those gaps. While the specific deliverable (a planning document) for this milestone is to be completed in Q2 FY08, the petascale infrastructure components will likely be deployed over the next ten years, with prioritization given to enabling predictive weapons simulations. The plan will be applicable to multiple ASC petascale platforms deployed during that time, including Roadrunner, the Sequoia Initial Delivery (ID) and Sequoia final systems.

2 ORGANIZATION AND CONTENT

This section contains information and excerpts from several previously published (or internal) ASC planning documents and provides a brief overview of the ASC platform strategy, system characteristics of concern for a balanced infrastructure, some user and application perspectives on petascale computing, and a short discussion of a proposed Usage Model for ASC Petascale Computing.

Following this section are four specific technical area discussions: Development Environment and Tools; Petascale Data Analysis; I/O, File Systems, and Archives; and Networks and Interconnects. Each technical section provides a brief overview followed by a description of major areas of concern. Each concern is addressed by a strategy and characterized by priority, difficulty, and cost. A timeline is provided at the end of each specific technical area section that states when efforts of the various strategies will result in deployable infrastructure components.

This document uses a “phased approach” where research, development, and deployment of infrastructure components are tied to one or more time frames to show how and when CSSE/FOUS infrastructures support the ASC program’s petascale acquisition and deployment planning. Three phases are defined in this document:

- Phase I, roughly FY08–FY10, represents a shorter term time frame and includes the early deployment of the LANL Roadrunner system and the LLNL Sequoia ID system.
- Phase II, roughly FY11–FY13, represents a medium term period that includes the LLNL Sequoia Final Delivery platform and possible petascale follow-on capability system(s) to ASC Purple.
- Phase III, roughly FY14–FY16, represents a longer term strategic view for future petascale architectures that are beyond the ASC program’s current platform planning, and perhaps beyond current conventional thinking.

In addition to phasing and scheduling, technical approaches are prioritized in "importance" and characterized as to levels of "difficulty" and "cost."

Three priorities will be defined:

- Essential: indispensable for a petascale system to function at the most basic level.
- Highly Important: necessary for system scalability reasons or required user productivity.
- Important: considered highly desirable for applications or users to achieve full system performance or scalability.

Three levels of difficulty will be defined:

- Hard: multiple person-years of effort with high potential for unforeseen technical challenges and no guarantee for success.
- Medium: multiple person-years of effort, but technically straightforward.
- Easy: Technically straightforward, but may require multi-site or multi-agency agreement and coordination.

Three levels of cost will be defined:

- $$$: multimillion dollar effort for lab labor and/or large external contracts.
- $$: efforts expected to cost over $1M.
- $: efforts expected to cost under $1M.

At the end of the document, a summary and some general conclusions are provided, including a description of overall CSSE/FOUS development and deployment strategies, a discussion of some areas of technical concern crosscutting more than one technical area, a consolidated timeline for component deployments and their relationship to ASC platforms, and some observations about future ASC petascale and exascale computational environments and infrastructure.

3 ASC PLATFORM STRATEGY

The ASC program recently published an *A Platform Strategy for the Advances Simulation and Computing Program* (*ASC Platform Strategy*). Computational platforms are an essential part of the tool set that ASC makes available to the weapons physics and engineering communities. While it is possible to run all problems on very expensive petascale computers, it is not the best use of resources. Four major principles guide the overall strategy for acquiring platforms to meet mission need as the Program balances needs and resources for solving today's problems while providing for more productive and cost-effective platforms for future problems.

Maintain Continuity of Production

It is necessary to maintain the productivity of the code developers and designers by providing architectures that, although they make increased computing power available, do not require all weapons work to slow while the codes are rewritten and ported to the new machines. This principle implies a conscious choice of continuity of infrastructure so that work can continue uninterrupted.

Ensure the Needs of the Current and Future Stockpile Are Met
Two realities drive us to focus on the future while committing to get the job done in the present. One is that the complexity of the simulations we need to run is increasing as we transition from ad hoc, model-based, calibrated codes to ab initio, physics-based codes. The other is that the supercomputing technology continues to evolve at a rapid pace. Together, these factors lead to the conclusion that future simulations are likely to be much different from those of today, and the ASC program must strike a balance between making investments to meet current mission workloads and the imperative to be prepared for tomorrow's mission workloads.

Balance Investments in System Cost-Performance Types with Computational Requirements
Capability, capacity, and advanced systems offer a range of capabilities and costs to the Program. Simulations capitalize on the features offered by each at different costs. The ASC program must invest in cost-efficient system types to match workload demands.

Partner to Introduce High-End Technology Constrained By Life-Cycle Cost
The Program must motivate industry to provide much increased capability and, as appropriate, drive the technology into new, promising, and applicable directions that have the potential to decrease time to solution and increase productivity by several orders of magnitude. However, the industry is now capable of building systems at scales beyond the reach of most operating budgets. The Program will need to work with vendors to ensure that hardware designs take into account operating costs.

The above principles will guide capital investments for both production petascale systems that maximize current productivity and in advanced petascale systems that are focused on future productivity improvements. Applying these principles moves platform (and infrastructure) acquisitions along parallel paths: we acquire incremental processing and memory improvements to our production capacity and capability platforms, and we work with industry to develop advanced systems with the necessary potential to improve productivity and/or reduce operating costs.

4 SYSTEM AND INFRASTRUCTURE CONCERNS

System and infrastructure considerations for future, well-balanced petascale environments are described in the *ASC Platform Strategy* appendix. A summary is presented here as background to support infrastructure component strategies and alternatives found in the technical report sections.

Processors
Over the time frame of interest for petascale systems, Moore's law will continue to govern. Moore's law encapsulates the empirical observation that the number of transistors on a die at constant cost doubles about every twenty-four months. However, Moore's law is silent on transistor performance. In the past, shrinking transistor feature sizes permitted a drop in circuit voltage and enabled an increase in the core's (CPU's) frequency at a rate that led to application performance doubling about every eighteen months. However, CMOS feature sizes are now so small that continued frequency increases seriously increase power consumption. On the other hand, using smaller feature size to add more cores has little effect

on power consumption. The majority of expected future microprocessor performance boost will thus come from a geometric increase in cores and at a much diminished rate from core frequency increases. This means processors will increase from dual-core and quad-core today to 32−128 or even more cores.

Memory

Microprocessors with 32−128 cores (CPUs) will force microprocessor designers and system architects to address balance factors for processor access to local memory hierarchies. With this many cores per processor die, it will be easy for a large fraction of the total computational capability of the processor complex to lie idle while waiting for data, and will force an effort to address the memory wall again. Understanding balance between the need for larger and more capable memories and increased computational speed will be imperative for ASC codes to perform optimally. As we increase physical fidelity and detail in our codes, improve numerical algorithms, and increase resolution, the need for more memory as well as more capable memory will continue to grow. We need to understand what demands ASC applications continue to make on memory subsystems and what the balance between processing and memory should be.

Interconnects

Scalable system architectures will also drive requirements for improvements in the interconnect fabric to take advantage of potential memory subsystem improvements. Areas for improvement include interconnect bandwidth, latency and message injection rate. In addition, this time frame may also see the development and use of optical technologies for interconnect fabrics. Future scalable system interconnect bandwidth requirements will accelerate the practical viability (price) of optics for interconnect technologies. Additional drivers include the weight of copper cables and the imposition of shorter distance limits as signaling rates increase on copper cables.

Accelerators

A new development that could have significant impact on our ability to achieve exascale levels of performance before 2020 is incorporation of SIMD or vector accelerators on scalable system compute nodes. Achieving scalable performance on heterogeneous architectures will only be possible if the coupling between cores and accelerators is extremely tight. It will also require continuing focus of interdisciplinary efforts to develop a new generation of parallel algorithms and their associated advanced solvers that are able to circumvent the interconnect and memory subsystem bottlenecks between the compute nodes and their integrated accelerators. Recent work on Roadrunner is an example of current attempts to attach accelerators into nodes on large-scale systems.

Operating Systems

Scalable system software is a critical enabling technology for future systems. We expect to have full service operating system software like Linux as well as lightweight kernel (LWK) operating system software. ASC invested in two systems that use LWK system software, Red Storm/XT3 and BlueGene/L, and both have demonstrated scalable and reliable performance up to full scale. While pursuit of application performance at petascale and beyond may also continue to require the use of an LWK, there are users that want to use some functionality provided with a heavyweight operating system. There are efforts supporting a broad range of

approaches to span the gap between full-featured Linux operating systems and stripped down LWK operating systems and runtime system software. The move to a large number of cores may also require development of new ways to distribute workload.

File Systems
Scalable parallel file system technologies are critical enablers for petascale systems and also as an integrating element within a simulation environment of capacity and capability systems, data analysis engines, and archival systems. Looking forward, it appears the only way to achieve I/O performance targets for petascale systems is through larger aggregations of devices and links. A major stumbling block to such levels of performance is the required number of devices. Large numbers of component parts pose challenges for integrated system management, fault tolerance, tuning and diagnosis of performance issues. These technical challenges are analogous to those faced 15–20 years ago when the first MPP systems were developed. Bandwidth, reliability, and cost are all critical issues for transition to petascale.

Reliability, Availability, Serviceability
Reliability, availability, and serviceability (RAS) will need improvements in capability and functionality to support the ability to run millions of cores on a single large problem. We need the ability for integration and communication among the operating system, runtime system, application software, and parallel file system when failures occur. As noted above, component part count for parallel file systems may drastically increase for petascale systems. The overall system will have to be highly resilient to failure of components.

System Management and Monitoring
The process of failure detection, identification, and fix is time consuming. Future systems will need to monitor themselves, automatically identify typical failures, and initiate corrective action. This kind of “autonomic” behavior is essential to operate more hardware without hiring more staff. Work is starting on initial steps to integrate data feeds into common monitoring frameworks to speed problem detection and identification by making a relevant data set available to key people. New tools will be developed to track application level test results over time. In the future, monitoring must be enhanced with scripts capable of taking corrective action when frequently encountered problems are detected.

5 USER AND APPLICATION CONCERNS

The following application considerations are a summary of programming trends important for ASC petascale systems, and are presented here as background to support some of the infrastructure component strategies and alternatives found in the following sections.

Current Programming Models
Early on, ASC standardized on a programming model for distributed memory with an explicit MPI for parallel communication. The investment made in application software and associated algorithms that use single-program-multiple-data (SPMD) programming model has paid dividends in the ability to port ASC applications across most of our capability and capacity systems with relatively modest levels of effort. Some applications also utilize OpenMP and POSIX threads for SMP parallelism to create a hybrid multi-level SPMD model, but current ASC codes largely use only a single level of parallelism. Typically, data are decomposed and distributed across the system and the same execution image is started on all MPI processes and/or threads. Exchanges of remote data occur for the most part at regular

points in the execution, and all processes participate in each such exchange. Data are exchanged with individual MPI send-receive requests, but the exchange as a whole can be thought of as a "some-to-some" operation with the actual data transfer needs determined from the decomposition. It should be noted that many SSP applications currently require at least 1–2 GB of memory per MPI task.

Future Programming Models

As the industry moves to parallel applications at unprecedented levels, applications will need to explore departures from current programming models to address performance and scalability issues on advanced systems. The ASC program must decide whether to make changes to its applications portfolio to use new programming models that exploit the computing potential offered by an advanced architecture based on new multi-core processors and/or heterogeneous approaches. This decision will probably be driven by the need to improve parallel efficiency. Future ASC applications may use functional parallelism, but, if so, it will be in conjunction with an SPMD model for individual modules. Parallel I/O and visualization may use this approach with functional parallelism at a high level separating them from the physics simulation with SPMD parallelism within each subset.

Future Scaling Challenges

The challenge of how to scale ASC applications that use standard MPI messaging techniques to petascale multi-core architectures will be driven by the exponential increase in the number of cores per processing chip, by practical memory cost limitations, and by practical interconnect bandwidth and messaging rate implications. Key issues that will be considered during current and upcoming phases of ASC planning are how SMP-style parallelism, threading, speculative execution, memory latencies, transactional memory, or other architectural and code concerns may influence eventual solutions. Work is already ongoing to explore possible programming model options on the likely architectures envisioned for new ASC platforms.

Petascale User Perspectives

Specific user perspectives and issues for petascale were discussed at the initial February 2007 infrastructure meeting held in Las Vegas. It was pointed out that not all physics advances are available through simple brute force and that the next set of computational increments will probably be used to change scientific workflow and to improve basic physics models. Subsequent computational increments may heavily challenge current numerical methods, and architectural changes at petascale may force users to revisit "store/communicate/recompute" methodologies. Major changes in computational paradigms could lead to alternative solution techniques. Thus, appropriate infrastructure and tools will be needed to analyze algorithms and applications on new petascale architectures.

In the area of code development and tools, concerns included limiting cross-compiling, limiting the extent of platform-specific code rework, threading improvements, support for dynamically linked executables, scalable debuggers, and memory tools. For petascale data analysis, concerns included continued availability of commonly used visualization tools, resources for on-demand visualization, adequate disk space for petascale data sets, and Python for in-situ data analysis. A foremost concern is the need to focus too much attention on I/O. Users do not want to be forced into a hard-to-use collective I/O model. They would also like assistance with application I/O characterization and adequate capabilities for

massive visualization and restart files. For networks and interconnects, scalable collective operations are required, as is the ability to support different point-to-point communication topologies, and a low-memory-overhead MPI library.

In general, petascale architectures will be more complex than their predecessors and will create many new challenges for ASC and the HPC community. Petascale will undoubtedly exacerbate the "performance gaps" among individual system components and the applications, thus forcing new paradigms to merge traditional applications and petascale system perspectives. Some have suggested applying an "application-centric" approach for HPC, replacing the historical ASC "kiviat" diagram approach that balanced ASC platform system characteristics such as memory, disk, I/O rate, and archival capacities bandwidth. An application-oriented approach would attempt to balance MTTI/MTTF, archiving times, idle times, and restart overheads and dump times, to measure possible options against application requirements instead of system components.

Maintaining Critical Interfaces
As new systems are developed and deployed we must be careful to move forward critical system software infrastructure components that we depend on today and that must continue to work on new systems. Many of these components (PAPI, TAU, solver libraries, etc.) are open source efforts that are beyond the scope of a particular platform development but that users and higher-level tools depend on.

6 PETASCALE USAGE MODELS

Detailed ASC user needs and desired capabilities have been previously captured in the *ASC Computational Environment (ACE) Requirements*. Usage models were subsequently developed for the newest terascale platforms (Red Storm and Purple), and addressed relevant and critical requirements from the latest version of the *ACE Requirements*.

To focus on issues key to a successful user environment for petascale platforms in the 2009–2013 time frame, we document below some critical requirements and a potential petascale Tri-lab usage model. It includes specific capabilities, tools, and procedures to support both local and remote users, and is focused on the needs of the ASC user working in the secure computing environments at LANL, LLNL, and SNL.

Usage model capabilities to address ACE requirements on terascale platforms have been traditionally divided into the following high level sections:

- Getting started (learning about the system, gaining access).
- Setting up the work environment.
- I/O and data migration.
- Application and system code development.
- Problem setup.
- Running the application to solve the problem.
- Processing simulation output.
- Tri-lab coordinated operational support.

Each of the above sections is briefly examined with respect to only those requirements and capabilities of importance to petascale. Many issues are further detailed in the Technical Working Group sections of this document. The focus below is on those capabilities driven primarily by the requirements of a remote user.

The potential for greatly increased size of petascale application data, both in numbers and size of files, has impact on two areas in particular: I/O and data migration, and processing simulation output. In both areas, there are challenges in the local environment and additional challenges for remote users, including the potential for reduced performance and productivity. In addition to the computing platforms, an extensive infrastructure, including hardware and software, must be supported. For a remote user, computing-at-a-distance adds challenges to both hardware (networking and storage) and software (data analysis and visualization tools) that must be considered.

Getting Started

We assume there are no major petascale environment issues with documentation, training, consulting, and account and password management (authorization). There may be some system availability and scheduling issues related to major shifts in heterogeneous architectures and programming models. However, authentication is the one capability that will have a different impact on the remote versus local users of a petascale system. For example, the security model is often slightly different for a ‘local’ user who can access a data transfer tool directly, rather than through cross-domain security, when transferring data between sites.

Setting Up the Work Environment

There are no anticipated petascale environment issues in setting up paths, environment variables, user groups, modules, and file system usage as it relates to locations and naming conventions for home, directories, and projects.

I/O and Data Migration

As stated above, the potential for greatly increased scale of application data impacts this area. There will be challenges in the local environment as well as challenges for remote users. Data transfer tools must be improved to increase performance on larger aggregate data sets while still operating efficiently on small files (< 2 GB). The requirement for improvement will spike upward with the location of capability machines remotely for some users; metadata must be optimized for tens of thousands of files to reduce transfer latencies. In addition, data transfer performance between sites may be different for users from each site. In a petascale environment, complete data sets could be too large to move. That has several implications, including the possibility that the remote site will need to provide additional archival storage.

Application and System Code Development

For petascale, issues exist in all areas of application and system code development (uniqueness of systems, parallel programming models and runtime systems, third-party libraries and utilities, compilation, debugging and correctness testing; and performance measurement, analysis, and tuning). In general, issues are similar for local and remote users. However, some GUI-based tools may not work as well remotely. Performance tools that generate voluminous data may also cause data access and data transfer concerns.

Problem Setup
There are no anticipated petascale environment issues in problem setup beyond the increased scalability required for domain decomposition tools.

Running the Application to Solve the Problem
There are no anticipated petascale environment issues in submitting the job, monitoring job status, stopping the job, and interactive use (other than interactive visualization). However, adapting the job for expected system reliability on petascale architectures may be a concern. Solutions for making the application runs more fault resilient may create new issues in I/O, development tools, and problem setup.

Processing Simulation Output
Again, the potential for greatly increased scale of application data has direct impact. There will be challenges in the local environment as well as additional challenges for remote users. Current ability to analyze petabyte data sets in situ will become more difficult and less efficient for the remote user until technology and funding make it possible to move petabyte data sets between sites rapidly and often. With capability platforms located remotely for some users, data analysis tools must be able to process petabyte-scale data sets and operate at efficient interactive rates between sites.

7 UNIQUE POSITIONING

ASC occupies a unique position in the overall computer science community for several reasons, including a highly collaborative business model, a necessity for secure environments, and a proven track record of strategically successful technical accomplishment.

A history of close collaboration among the three primary DP laboratories as well as key industrial and academic organizations has yielded multiple payoffs. Collaboration leverages critical, and sometimes scarce, intellectual resources. Collaboration avoids redundant activities and improves resource utilization. CSSE/FOUS collaboration facilitates development and deployment of portable solutions that can be shared among multiple users. CSSE/FOUS efforts also support some of the nation's few classified supercomputing environments with facilities and experienced staff to support classified, as well as unclassified, stockpile stewardship computing.

CSSE/FOUS systems and products are widely regarded as trailblazers and have earned recognition and respect from peers and many R&D 100 Awards (HPSS, HDF5, Chromium, VisIt, Global-Link DVI over gigabit Ethernet, Science Appliance, Sapphire, encryption advances, and 10-gigabit Ethernet optimization). Technical successes are evident in other examples, including a history of being atop the Top500 list of world's fastest computers, the development of lightweight operating system kernels (Puma, Cougar, Catamount), driving the development of the TotalView scalable debugger, playing a critical role in the development of the OpenMP organization and its specifications, pioneering use of large scalable file systems (Lustre, Panasas), scalable, lightweight cluster management systems (CHAOS/TOSS), high-performance, fault-tolerant runtime libraries (LA-MPI/Open MPI),

3D immersive visualization environments, and some of the best examples of production large Linux clusters at scale for capacity computing and visualization.

As leading-edge national user facilities for high-end computing, a primary focus on NNSA stockpile stewardship goals is both appropriate and necessary. We have an exceptional track record in support of mission priorities and, in doing so, promote advancement of national supercomputing capabilities. NNSA cannot sit back and just procure the petascale infrastructure it needs from off-the-shelf sources. Due to (1) our unique requirements for many capacity computing platforms with a usage sweet spot four-to-eight times higher than typical commercial systems, (2) the need for petascale (capability) platforms for our most demanding simulations, and (3) problem-optimized systems (advanced architectures) aimed at solving specific outstanding weapons simulations problems, we must define and follow through on judicious computational strategies. Achieving our goals requires sustained investment in focused research, development, and deployment to ensure the technologies that address ASC's unique mission-driven need for scalability, parallelism, performance, and reliability. The petascale environment infrastructure strategy will likely continue to be "buying what we can" complemented by "developing what we must."

PROGRAMMING ENVIRONMENTS AND TOOLS

1 INTRODUCTION AND BACKGROUND

ASC codes represent a substantial investment of programmer effort. Even the youngest ASC codes have been in development since the beginning of the program and represent at least 10 years of multi-person effort; most have taken twenty or more years to develop. This substantial investment requires that we minimize the effort required for the codes to use petascale architectures, a fact that application developers invariably state. However, we anticipate significant changes in computer architectures: processor technology is rapidly changing as chip vendors experiment with increasing on-chip parallelism as a way to follow Moore's law. We are increasingly seeing processors with multiple cores and with vector units, and in the future we can expect to see graphical processing units (GPUs) and other hybrid devices integrated into commodity chips.

Our challenge is to provide an environment that supports efficient use of emerging petascale systems without requiring hundreds of person-years of new programming effort. We must provide new and innovative programming environments and tools to support code design, creation and modification, including building and debugging applications and tuning their performance. Each of these tasks involves significant complexity, particularly in the context of the multi-physics applications that characterize the ASC integrated codes.

2 DEVELOPMENT/DEPLOYMENT AREAS

Three important themes emerge as the key development areas for programming environments and tools. These themes are the consideration of the programming models needed to effectively program for new computer architectures, performance analysis tools to aid in achieving adequate performance, and correctness tools to ensure that applications provide the correct results. As petascale architectural directions evolve, ASC integrated codes must evolve with them, and this is likely to require an evolution in the programming models upon which the applications are based. Experience on terascale architectures has shown that large-scale systems pose unique challenges in correctly implementing algorithms that use the architectures efficiently, and an additional order of magnitude in system size and performance brings new challenges. Thus, we anticipate the need for both correctness tools and performance analysis tools that target petascale systems. Strong interactions exist throughout all three areas. For example, new programming models may require additional investments in correctness and performance analysis tools.

The following sections detail our anticipated concerns for each theme and our recommended strategies for alleviating them. We provide approximate time frames for aspects of each strategy; near term implies within two years, medium term implies between two to five years and long term implies between five to ten years. Many strategies have a range of aspects over all terms.

2.1 PROGRAMMING MODELS

Programming models, and the languages and the libraries that implement them, must adapt to the growing levels of on-chip parallelism and the increasing depth of memory hierarchies. Almost all ASC codes currently use the SPMD programming model with MPI for communication. This programming model has served ASC applications very well in cluster and SMP environments over the past ten to fifteen years. Unfortunately, the enormous levels of parallelism anticipated in

petascale systems are likely to make this MPI everywhere model insufficient. The anticipated massive scale of the new architectures also raises concerns regarding power usage and hardware failure rates. Thus, we must explore the use of new programming models. The significant time and investment required to create the current ASC applications implies that we must ensure that these new programming models do not require massive rewriting of this code base.

2.1.1 CONCERN: Lack of a Common Programming Model for New Architectures

Multicore and heterogeneous architectures are bringing about a renewed interest in parallel program models despite the lack of a widely accepted programming model beyond MPI. Any new programming models must address the issue of multiple layers of local and global communication. The long-term direction of processor architectures requires solving this issue to ensure that codes can run efficiently on all ASC platforms. We anticipate that message passing will meet petascale requirements for inter-node communication, but that intra-node programming and communication may require an additional solution. Advances in compiler technology alone may help, but probably only in the long term. OpenMP provides an example of a promising approach: mechanisms for annotating code to provide guidance to the compiler and an organization of all major hardware vendors working towards a common programming model solution. However, currently OpenMP overheads are too high for ASC codes to achieve adequate performance. Furthermore, it does not support memory placement for NUMA systems adequately, let alone for explicit memory architectures such as Cell-based systems.

2.1.1.1 STRATEGY: Investigate and Develop New Programming Models

Priority: Essential **Difficulty: Hard** **Cost: $$**

While hardware directions may reduce OpenMP overheads, we must investigate how additional code annotations or other programming paradigms can facilitate exploitation of fine-scale parallelism in representative ASC applications on chip architectures upon which petascale platforms are likely to be based. Thus, our near-term strategy is to examine OpenMP and various threading models, streaming, data parallel, functional and proprietary approaches in light of an understanding of the ASC codes performance characteristics in order to develop an API that can be specialized to meet ASC needs. The initial process, which will continue over the medium term, will provide optimization benefits to the core code base as the programming model develops. For our long term-strategy, we will work with the OpenMP community and others to ensure that promising solutions become standardized. Throughout this strategy, we will work directly within existing ASC applications to optimize them for petascale systems, both to guide emerging programming model directions and to incorporate new programming model solutions.

2.1.2 CONCERN: Lack of Mechanisms to Control Data, Thread, and Task Placement

Performance on petascale systems will strongly reflect memory and communication locality issues and questions of where specific code segments are run, whether on an accelerator, or on which core of a multicore chip (i.e., code placement issues). Until a compiler-based solution (like OpenMP) has demonstrated success in achieving adequate performance on petascale architectures, mechanisms must be developed that give the programmer explicit control of memory and thread placement. The ability to manage memory usage and locality efficiently is essential for using petascale architectures effectively. APIs and implementing libraries are required that expose the memory and network topology of the hardware and give the developer explicit control of communication and fine-scale thread synchronization. In the least, code

placement issues will be relevant to all potential Tri-lab petascale platforms. Currently, no standards exist for explicit code placement and memory usage, without which low-level, platform-specific coding may be required. While such standards are clearly needed in the long term, portable libraries can more quickly provide a solution that allows for easier transition of an application from one architecture to another while effectively using the memory model or communication topology employed by the architecture and those needed by the application.

2.1.2.1 STRATEGY: Develop Library Implementations for Data, Thread and Task Placement

Priority: Highly Important **Difficulty: Medium** **Cost: $$**

Because of the overarching importance of data locality and data movement in scientific application performance, we recommend near-term experimentation with and refinement of APIs and libraries for explicitly programming threads and memory topologies. These APIs must allow the association of software threads with hardware threads and their local stores and with data transport between threads. In the same time frame, we should create tools that inform the user of the inter-node communication topology and associated latencies and allow the placement of code across nodes in a manner that takes into consideration the discovered topology. We anticipate these investigations leading to production quality solutions for specific platforms over the medium term. The long-term need is for portable, standardized solutions, which requires close cooperation with hardware and compiler vendors and academic partners exploring streaming and other models and may best be folded into the programming model standardization effort.

2.1.3 CONCERN: Lack of Tools to Support the Migration of Existing Applications

The transformation of existing codes to achieve adequate performance on petascale hardware architectures is a daunting task. Currently, few tools exist to aid the programmer in making this transition. In addition, a path is needed for existing codes (especially Fortran) to participate in any new programming models that arise. Compiler and support tools are needed to enhance the capability to optimize or to transform applications to use different architectures effectively, with less impact on the developer and the code base. Existing Tri-lab C and C++ compiler technology can provide tools to help in the migration of existing codes. Very recently, this technology has served as the basis of a new class of tools that automatically transforms data structures to a format that provides the most efficient use of the memory system of a specific platform. These tools require straightforward changes to the application code once, after which we only need to capture specific aspects of the architecture once for all applications that have made the required changes. However, a significant concern with this possible direction is that compiler infrastructures are complex and difficult to maintain, while ASC application teams require robust and sustainable solutions.

2.1.3.1 STRATEGY: Develop Tools to Aid in the Transformation of ASC Applications

Priority: Highly Important **Difficulty: Hard** **Cost: $$$**

We must continue to support and to expand the open-source compiler technologies that enable the creation of these powerful source-to-source transformations. The near-term goals will be to provide Fortran support in this infrastructure and to generate requirements for additional tools that reduce the programmer effort required to migrate existing applications to petascale platforms. More importantly, we also must continue current ASC efforts in source-to-source transformation frameworks in order to ensure the needed robustness and sustainability. In the medium term, we must spawn partner-supported efforts to develop the compilation-based tools

to support pre-compiling, source-to-source transformations and enhancements to existing compilers. The long term strategy must continue this research because it will provide the flexibility needed to support the scale and potentially heterogeneous nature of future platforms.

2.1.4 CONCERN: Lack of Mechanisms to Handle Soft Hardware Error Conditions

Currently, no standard mechanism for the reporting of soft hardware errors to the application exists, nor does a standard for error recovery beyond checkpointing and restarting. As hardware error probabilities increase (because of higher component count), the more critical it is to address this gap. Reducing the cost of checkpointing can help; however, programming tools must support responses beyond simply restarting, since error rates could become extremely frequent with petascale node counts. Incorporating fault tolerance into the algorithms and programming models used by ASC applications would increase resource utilization of all ASC platforms.

2.1.4.1 STRATEGY: Develop APIs and Mechanisms to Compensate for Hardware Errors

Priority: Important Difficulty: Low (near)/Hard (long) Cost: $ (near)/$$ (long)

Because we anticipate the importance of this concern will increase over time, we propose an evolving strategy. In the near term, we must explore APIs and develop runtime tools that monitor the state of hardware and report on error conditions such as component overheating and throttling down. These relatively simple goals can be accomplished at low cost in a two-year time frame. Over the medium term, we will explore mechanisms to make the runtime environment, including MPI, reconfigurable so that problem nodes can be isolated and an application load balanced for the new configuration. We expect it to take two to four years to provide initial implementations of these short- and medium-term solutions; development of these solutions should begin no later than late FY09. The long-term strategy is far more complex: we must develop algorithms and programming models that use these new APIs to detect problems automatically and to compensate for them while minimizing disruption of the running job to enhance overall throughput of capability platforms significantly. This strategy will require coordination with operating system and resource manager implementers.

2.1.5 CONCERN: Extreme Power Consumption of Petascale Systems

Current large-scale systems consume 5 MW of power or more, which equates to about five million dollars per year. Scaling these systems to the petascale would require prohibitive power costs. While most petascale designs promise to reduce that target, very high power consumption at peak operating conditions is still likely. Recent research has explored using mechanisms like dynamic voltage scaling to achieve equivalent performance with significant power savings. Software mechanisms to reduce power consumption, while not increasing time to solution, would either reduce programmatic costs or accelerate completion of mission critical activities.

2.1.5.1 STRATEGY: Develop User-Level APIs and Mechanisms to Reduce Power Usage

Priority: Highly Important Difficulty: Low (near)/Hard (long) Cost: $ (near)/$$ (long)

We also propose an evolving strategy to address the rising cost of running ASC capability calculations. In the near term, efforts should be initiated to work with vendors and Tri-lab operating system implementers to develop mechanisms to allow user-level control of voltage scaling. Power costs motivate further investigation of these approaches, despite not directly impacting programmers or applications. We will continue initial small activities investigating

techniques to provide power efficiency without sacrificing performance in MPI and OpenMP programs in the near term. If these initial activities provide encouraging results, consideration should be given to starting larger efforts to research directions to automate power aware application-level techniques and to develop production solutions starting in FY10 or FY11.

2.2 CORRECTNESS TOOLS

Our correctness tools thematic area addresses the tools that developers use to ensure that programs run to completion and give the expected result. The needed tool set includes traditional debuggers that allow programmers to set breakpoints, to run, to step though, and to examine and to modify data in a running program. However, the scalability of this approach is unlikely to extend to full petascale systems. Further, these traditional tools often provide too little insight into the root causes of errors. Thus, we propose an overall correctness tool strategy that combines these traditional debuggers with lightweight tools that provide critical, but limited, debugging capabilities at large scales, while providing more scalability than traditional debuggers and with automated correctness tools that can automatically provide greater insight into root causes based on semantic knowledge of certain aspects of the underlying programming methodologies.

2.2.1 CONCERN: Traditional Debuggers Do Not Scale to the Level That Code Teams Need

Traditional debuggers allow programmers to manipulate a running application to understand and to correct program behavior. Currently, application programmers find traditional debugger performance limiting beyond a thousand processors (or fewer) and the time for typical operations increases exponentially beyond four thousand processors, with even the most scalable debuggers. Performance of traditional debuggers must handle reasonably large job sizes even when combined with lightweight strategies to narrow the problem space. Further, few lightweight tools exist, and forming our petascale debugging strategy solely around the expectation that production quality solutions will emerge in the near and medium term in which they are needed could leave us with no viable debugging strategy for the most important petascale application runs.

2.2.1.1 STRATEGY: Work with Suppliers to Improve Scalability of Traditional Debuggers

Priority: Essential **Difficulty: Medium** **Cost: $$**

In the near term, we must work with implementers of existing traditional debuggers to deliver tools that can handle users' current needs to debug at scales greater than a thousand MPI tasks. Further, users need tools in the near term that allow them to handle even larger task counts on petascale systems while lightweight solutions are developed. Combined with users' existing familiarity with traditional debuggers, we must pursue a near- and medium-term strategy that improves their scalability and ensures that they can extend to at least eight to ten thousand MPI tasks. In the long term, we must assess our lightweight debugging solutions, which may make traditional debugger scalability improvements beyond eight to ten thousand tasks unnecessary. In addition, the overall petascale debugging strategy requires that subset attach mechanisms work on petascale systems, which we can ensure by working closely with platform vendors from near to long term.

2.2.2 CONCERN: Traditional Debugging Paradigm Does Not Scale to Petascale Systems

As discussed above, traditional debuggers do not scale to the level needed for petascale systems. While additional investment can alleviate this problem, these tools use mechanisms and display techniques that will not provide sufficient usability at millions, or even tens of thousands, of MPI tasks. Specifically, setting breakpoints or stepping through individual lines of codes at this level is unlikely to proceed in reasonable time regardless of how much money we invest in traditional debuggers. Similarly, no user can sort through the displays of this many execution contexts to determine which ones are exhibiting errors. Thus, our overall petascale debugging strategy includes lightweight tools that narrow the number of contexts to which we apply traditional debuggers. However, we currently have no lightweight debugging tools deployed for production use.

2.2.2.1 STRATEGY: Complement Traditional Approach with Lightweight Debugging Tools

Priority: Essential **Difficulty: Medium** **Cost: $$$**

Initial work in these directions indicates that we have the appropriate expertise to design and to build lightweight debugging tools. We must aggressively continue these efforts to improve our ability to narrow problematic code regions at scale throughout the time frame addressed by this report. This will require significant near-, medium-, and long-term development effort to implement production ready versions of existing prototypes as well as additional research to improve our ability to identify root causes. Further, we neither have nor anticipate adequate staffing to provide support and maintenance of many production-quality tools in-house. Thus, as solutions are identified, we must not only work to develop them into high-quality tools but also to identify potential vendor partners (including ISVs) to handle the on going maintenance and support effort. This requires a near- to long-term effort to develop a broader market for these tools, including in Office of Science and DOD HPC centers.

2.2.3 CONCERN: Manual Identification of Parallelization and Other Errors Is Too Difficult

Applications will become more complex in order to accommodate the petascale architectures and to add more physics capabilities made possible by the new machines. Parallelization errors, such as deadlocks in message passing operations, or race conditions in threaded programs are not easily analyzed with traditional debugging techniques. Thus, a concern is that significant productivity losses will continue due to the need to identify the root causes of errors manually. Parallelization errors are similar to memory leaks and other memory access errors, such as accessing uninitialized memory. Existing thread correctness checkers are neither multi-platform nor designed for large-scale environments, while MPI correctness checkers either have similar limitations or are not robust enough for production environments. While we have developed automated tools to detect memory access errors, it is not clear that they will be available for petascale platforms.

2.2.3.1 STRATEGY: Develop Static and Dynamic Automatic Correctness Checkers

Priority: Highly Important **Difficulty: Medium** **Cost: $$**

Our overall petascale debugging strategy includes two complementary approaches to detect the root causes of errors automatically: static analysis to identify problems in source code, and runtime analysis to find additional incorrect programming constructs. In the near term, we must

adapt existing memory correctness tools to provide runtime thread-safety checks. Further, we must also accelerate efforts to develop and to improve the portability of existing static and dynamic correctness checkers, particularly for MPI. Over the medium term, we must work with third-party vendors to productize the solutions that we develop for the full range of petascale architectures. Throughout the time frame covered by this document, we must ensure that memory correctness tools area available on petascale systems, and we must support existing compiler infrastructures in order to provide static analysis capabilities needed by program correctness checkers.

2.2.4 CONCERN: Future Systems Will Have Significantly Less Memory Per Core

The expected trend of less memory per processor in petascale systems implies a significant change in the programming assumptions for ASC applications in order for their memory usage to scale across expected petascale node counts. Specifically, ASC applications must use less memory per MPI task in order to run on anticipated petascale systems. However, we currently have no tools to track the amount of memory allocated per call site or package accurately, let alone to capture allocation scaling trends. Even worse, we have no way to assess whether allocated memory is accessed or at what frequency, which means we cannot assess if the cost of computing the stored data justifies using the limited main memory resource. Overall, our concern is that ASC codes will be unable to use petascale systems due to poor memory usage scaling, or productivity losses due to inadequate memory usage scaling tools will prevent achieving important programmatic milestones.

2.2.4.1 STRATEGY: Develop Mechanisms to Analyze and to Reduce Memory Usage

Priority: Essential **Difficulty: Medium** **Cost: $$**

In the near term, we will develop in-house tools that track the size of memory allocations associated with source code lines and routines. We will augment these tools in the same time frame with mechanisms to extrapolate scaling behavior and to assist users in identifying key trends for effective use of petascale systems. We need these tools immediately as memory usage scaling already impacts integrated codes on terascale architectures. Proper investment should lead to production solutions by early FY10 that only require porting efforts thereafter. In the near to medium term we will investigate approaches to help users decide if recomputation or out-of-core storage to save memory space would be advantageous. If these research efforts result in promising approaches, they may merit additional investments; we will evaluate this question as the program progresses.

2.2.5 CONCERN: No Debugging Solution Exists for Heterogeneous Systems

The adoption of heterogeneous chip designs holds the potential for dramatic gains in application performance. However, heterogeneous architectures are very new (or still in the development stage) and debuggers for them do not currently exist. Code development on heterogeneous architectures is just now beginning, and developers are unfamiliar with these new architectures. Thus, debugging facilities are even more critical on these new machines. Full-scale debugging support on heterogeneous machines may not be developed in the time frame needed for planned systems, and even printing from a specific hardware thread may be problematic. In addition, hardware vendors may supply a chip-level debugger, but the financial incentive may not exist for a vendor to provide for debugging across multiple heterogeneous nodes.

2.2.5.1 STRATEGY: Implement Heterogeneous Debugging Tools

Priority: Essential **Difficulty: Medium** **Cost: $$**

We must initiate efforts with vendors in the near term to provide for the debugging of applications running on heterogeneous systems. Given the considerable risk in relying on vendors to provide a multi-node solution, we will also pursue an open-source strategy for heterogeneous debugging. This effort should begin in the FY08 time frame and will likely take two years to accomplish.

2.3 PERFORMANCE ANALYSIS TOOLS

Using terascale architectures efficiently has proven challenging. Frequently, integrated applications achieve a very modest percentage of the peak performance of the architecture. Our experience has demonstrated that this fact does not necessarily indicate that the system is not being used well. For example, the application may need the large number of nodes in order to have sufficient physical memory to run the problems of interest. Similarly, the application may be running at the limits of the node's memory bandwidth or latency. Ultimately, the limiting factor on an application's performance is a complex issue on large-scale systems that promises to become even more difficult in the petascale era. Not only do we need to improve our current capabilities, but we expect new architectures to imply new performance issues. The overall risk in this area is that applications will not perform sufficiently to solve mission critical problems in the time available. For this reason, we must continue to invest in performance analysis tools.

2.3.1 CONCERN: Load Balance Issues Will Prevent Significant Performance Gains

Load balance will be the single most important scaling issue in systems with hundreds of thousands or even millions of nodes. As system sizes in terms of number of cores continue to grow, small perturbations in the amount of computation per core will imply significant lost performance opportunity. While existing tools provide reasonable ability to measure the amount of computation per node, they do not provide sufficient insight into why imbalances occur. The root cause of the imbalance can be something as simple as having more particles assigned to some nodes. However, it is often something more obscure, such as small differences in memory allocations or initial state of the nodes. Our experience on relatively modest numbers of nodes indicates that merely measuring load imbalance is not sufficient, and it is unlikely that more advanced mechanisms will be developed without targeted investments. Overall, our concern is that minor imbalances on petascale systems will dramatically reduce performance and prevent the meeting of programmatic goals despite codes with high performance within computationally intensive regions.

2.3.1.1 STRATEGY: Develop Automated Mechanisms to Identify Load Balance Root Causes

Priority: Highly Important **Difficulty: High** **Cost: $$$**

In the near term, we must develop accurate mechanisms to measure load balance and to identify where imbalances impact performance significantly. These mechanisms must scale to full petascale system sizes and must neither perturb application performance significantly nor introduce spurious imbalances, requirements that will tax our tool infrastructures. Properly designed, they will not only improve application performance but will also enable hardware problem identification. In the medium term, we must deliver these methods in robust and

scalable tools and continue research to develop mechanisms that identify the underlying cause of observed imbalances. The long-term strategy must include continued productization as well as research into automated load balancing techniques. We expect that further refinements in our tools will improve our ability to identify root causes of load imbalances throughout the time frame of the requirements discussed in this report.

2.3.2 CONCERN: Tools to Assess and to Improve Data, Thread, and Task Placement Are Inadequate

Incorrect task placement decisions can reduce performance by a factor of two or more on BlueGene/L. The impact of poor code or data placement decisions on NUMA systems includes significant performance variability as well as performance reductions of 50% or more. Poor code placement decisions on hybrid systems could have even greater impact. We expect significant differences between MPI task placement, thread placement on NUMA systems, and code placement on hybrid architectures. Different techniques may even be required within one of these categories. Tools that assist programmers make these decisions could significantly improve time to solution. However, no significant production solutions currently exist to assess options for any of the various placement scenarios.

2.3.2.1 STRATEGY: Explore New Placement Tools and Productize Existing Tools

Priority: Highly Important **Difficulty: Medium** **Cost: $$**

We must continue efforts to understand communication locality, data and code placement issues. Because code placement is the overriding factor in performance of hybrid architectures, we must accelerate efforts to provide a near-term prototype solution with medium-term productization. For NUMA systems, a near- to medium-term time frame to develop prototype solutions that assess the impact of data placement should be sufficient with, productization following late in the medium term. We may be able to defer MPI task placement questions depending on the compute network architectures chosen. Thus, the near- and-medium term strategy for MPI task placement issues is to pursue small research efforts and to assess the need for accelerated efforts based on emerging trends in petascale systems.

2.3.3 CONCERN: Inadequate Memory System Performance Analysis Tools

While not unique to petascale systems, understanding memory system performance is essential to the effective use of modern computer architectures. Existing tools are inadequate for understanding memory system performance, which dominates single node performance on current architectures. Architecture trends indicate that this aspect will become more difficult due to a wider variety of memory architectures, including not only complex, multi-level, cache-based memory hierarchies but also other memory architectures that require the programmer to manage the hierarchy explicitly. However, some of these trends may make memory performance analysis simpler and we expect that ultimately all systems will adopt strategies that allow memory management to occur automatically from the application programmer’s viewpoint. We also anticipate vendors will develop tools to address this issue for commodity systems. Nonetheless, maximum performance is likely to require some programmer intervention for memory bandwidth intensive or latency critical code regions, and vendor tools may not capture important aspects of the problem specific to scientific computation.

2.3.3.1 STRATEGY: Continue Research into Memory System Performance

Priority: Important **Difficulty: Medium** **Cost: $**

Throughout the time frame of this document, we will continue to research mechanisms to understand memory system performance and to monitor directions in memory architectures. We will also evaluate vendor memory performance tool offerings. We will work directly with code teams to understand if acceleration of tool development in this area is required. This strategy will require modest investments in the near term. We will need flexibility to increase this funding quickly if we observe memory performance shortcomings for selected petascale architectures.

2.3.4 CONCERN: Possible Differences between Expected and Actual Performance

It is vital that the processing capability of each new system deployment be quantified in advance for the workload that will utilize it. Unless we continue to improve existing performance modeling efforts, it is likely that the expected performance will be inaccurate. Our experience has shown that a range of performance modeling techniques can aid in the early design of large-scale systems, in procurement to compare system proposals from multiple vendors, to verify performance during installation, and to assist in both software (algorithmic) and hardware optimization processes. In addition, with the increasing complexity posed by multi-core and heterogeneous processors, we will be unable to quantify the performance impact of software optimizations prior to their availability on the target system without refined advanced performance modeling techniques.

2.3.4.1 STRATEGY: Develop Improved Application Performance Prediction Capability

Priority: Important **Difficulty: Medium** **Cost: $$**

Building on current capabilities, we must develop and refine performance models that automate incorporation of developments in application coding, as well as innovations in the hardware architectures. Many of our current modeling capabilities have focused on the exploration of scalability and analysis at the large-scale, which must continue for petascale systems. These have been highly successful and have enabled a multitude of systems to be compared against those that have been deployed. However, we also note that achievable application performance is split across both the single-core/single socket performance and the effects of scaling. Thus, a near-term activity will develop modeling capabilities that enable a full spectrum of performance investigations as the number and type of cores on a chip increases. Over the long term, this activity will bring together several research activities in performance modeling that will cumulate for the analysis of achievable application performance to be investigated prior-to, during, and after system deployment for a multitude of hardware architectures and workload configurations.

2.3.5 CONCERN: Tool Infrastructure Development and Maintenance Not Valued Properly

Experience has shown that performance analysis tool needs are difficult to anticipate and are often unique to a given architecture/application combination. Further, uncertainty in petascale architecture directions is likely to lead to multiple architecture types (at least initially), and the required measurements and analysis for each application often has unique aspects. Thus, we need flexible, portable, and adaptable performance analysis tools in order for applications to use the systems effectively. A tool strategy that relies on a flexible infrastructure based on portable and

scalable modules, such as PAPI, will enable rapid response to these unpredictable needs. The infrastructure must include highly scalable communication, data reduction, data analysis, and data presentation mechanisms. Ongoing efforts to modularize existing infrastructure such as Dyninst will help satisfy this concern. However, those efforts may flounder without ASC investment or fail to achieve sufficient quality for our systems. Also, existing successes such as PAPI may not be supported on new architectures, such as the Cell, that may require new hardware abstractions. In any event, existing infrastructure solutions are often not of production quality and do not provide the full range of functionality, such as scalable tool daemon launch and attach mechanisms. Also, development of modular tool infrastructure by a wide variety of implementers and research groups could lead to interoperatability problems or fail to meet infrastructure performance needs. For example, the infrastructure is unlikely to scale to the anticipated sizes of petascale systems without special attention and targeted investment.

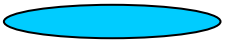
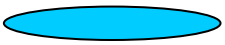
2.3.5.1 STRATEGY: Develop and Maintain a Scalable Community Tools Infrastructure

Priority: Essential **Difficulty: Medium** **Cost: $$**

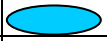
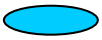
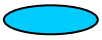
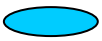
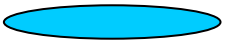
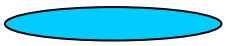
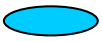
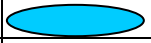
We must develop a robust, scalable, and flexible infrastructure that enables rapid development of application-specific tools. Because initial tool deployments on petascale systems should use this infrastructure to ensure that it functions on them properly, the development is a near-term facet of our overall tool strategy, which requires immediate investment. A flexible infrastructure for MPI tools is needed no later than late FY09; the time frame for scalable communication and job launch and control mechanisms is similar. We anticipate needs for infrastructure refinements to continue through the medium term.

3 TIMELINE SUMMARY

Our programming environment and tools strategies encompass continued research and development efforts in the areas of programming models, correctness tools, and performance analysis tools. In the near term (within the next two years), we must identify and refine the programming models to be employed on petascale systems. In that same time frame, we must implement the initial portions of an overall petascale debugging strategy and provide a scalable and robust infrastructure on which to build performance analysis tools as well as some aspects of the debugging strategy. Several other near-term activities are expected to lead to production quality solutions in the medium term (between two to five years). Finally, we have discussed several long-term directions, many of which include reassessing needs throughout our efforts to develop our usable petascale environment. The following table summarizes the timeline of all activities in this technical area.

Strategy	Activity	Near Term	Medium Term	Long Term	Comments
2.1.1.1	Examine OpenMP and other threading models, streaming, data parallel, and proprietary approaches				
2.1.1.1	Develop optimizations for ASC codes to guide and to reflect evolving programming models				
2.1.1.1	Work with OpenMP community and others to standardize promising approaches				

Strategy	Activity	Near Term	Medium Term	Long Term	Comments
2.1.2.1	Experiment with APIs for explicitly programming threads and memory				
2.1.2.1	Develop production quality tools to automate code placement				
2.1.2.1	Standardize solutions for explicitly programming threads and memory and automating code placement				
2.1.3.1	Develop Fortran support in source-to-source translation infrastructure				
2.1.3.1	Maintain source-to-source translation capabilities, research applications of it, and work with vendors to integrate into production tool set				
2.1.4.1	Develop APIs and tools to monitor and to report hardware errors				
2.1.4.1	Investigate reconfigurable runtime environments to respond to errors				
2.1.4.1	Develop algorithms and programming models that automatically detect error conditions and minimize their impact on running jobs				
2.1.5.1	Develop user-level mechanisms to control dynamic voltage scaling				
2.1.5.1	Investigate performance preserving power aware techniques and evaluate appropriateness of longer-term effort				
2.2.1.1	Extend traditional debugging capability to 4,000 MPI tasks				
2.2.1.1	Extend traditional debugging scalability 10,000 MPI tasks				
2.2.1.1	Work with vendors to ensure subset attach works on all platforms				
2.2.2.1	Identify and prototype lightweight debugging tools				Deliver useful tools throughout
2.2.2.1	Work with vendors and other HPC programs to productize lightweight debugging tools				
2.2.3.1	Adapt existing memory checkers to provide thread-safety checks				
2.2.3.1	Develop portable static and dynamic correctness checkers				Particular near-term MPI focus
2.2.3.1	Productize correctness checkers				
2.2.3.1	Support correctness checker infrastructure and porting efforts				
2.2.4.1	Develop tools to track and to predict scaling of memory allocations				Very near term
2.2.4.1	Productize memory allocation tool				
2.2.4.1	Explore tools to assess tradeoff between memory and recomputation				Evaluate possible long-term benefits

Strategy	Activity	Near Term	Medium Term	Long Term	Comments
2.2.5.1	Develop heterogeneous debugger				
2.3.1.1	Develop mechanisms to measure load imbalances and their impact				
2.3.1.1	Research automated load balancing techniques and productize solutions				
2.3.2.1	Develop code placement techniques for heterogeneous systems				
2.3.2.1	Productize heterogeneous code placement techniques				
2.3.2.1	Develop code placement techniques for NUMA systems				
2.3.2.1	Productize NUMA code placement mechanisms				
2.3.2.1	Research MPI code placement techniques and assess need				Accelerate if appropriate for selected systems
2.3.3.1	Continue to research memory system performance issues and to monitor memory system directions				
2.3.4.1	Develop multicore system performance modeling techniques				
2.3.4.1	Continue performance modeling research				
2.3.5.1	Develop flexible MPI tool infrastructure				
2.3.5.1	Develop scalable tool communication and job control mechanisms				
2.3.5.1	Port, maintain, productize, and extend scalable tool infrastructure				

PETASCALE DATA ANALYSIS

1 INTRODUCTION AND BACKGROUND

Until recently, the central challenge for ASC data analysis was to develop technology that enabled visualization and exploration of large data. This resulted in the development of distributed memory software that ran on commodity clusters that shared the work of reading and operating on large data. This was largely a post-processing effort, enabling interactive visualization and exploration of data read from disk after the simulation had completed. This approach has been successful for the Tri-labs, and with continued investment, we expect it to address basic needs for interactive petascale data analysis.

However, this post-processing approach to data analysis is I/O bound. The richness of the resulting data is limited by the speed with which data can be written to disk and overall capacity constraints—how much data are we able to store. Thus, discovery is limited by the fidelity of the data that can be written. It is important to note that in the absence of a better solution, customers resort to ad hoc methods of balancing the data that is written against the time that they have on the machine. Improving the data that is written to disk is a primary motivator of the forward-looking elements of this section.

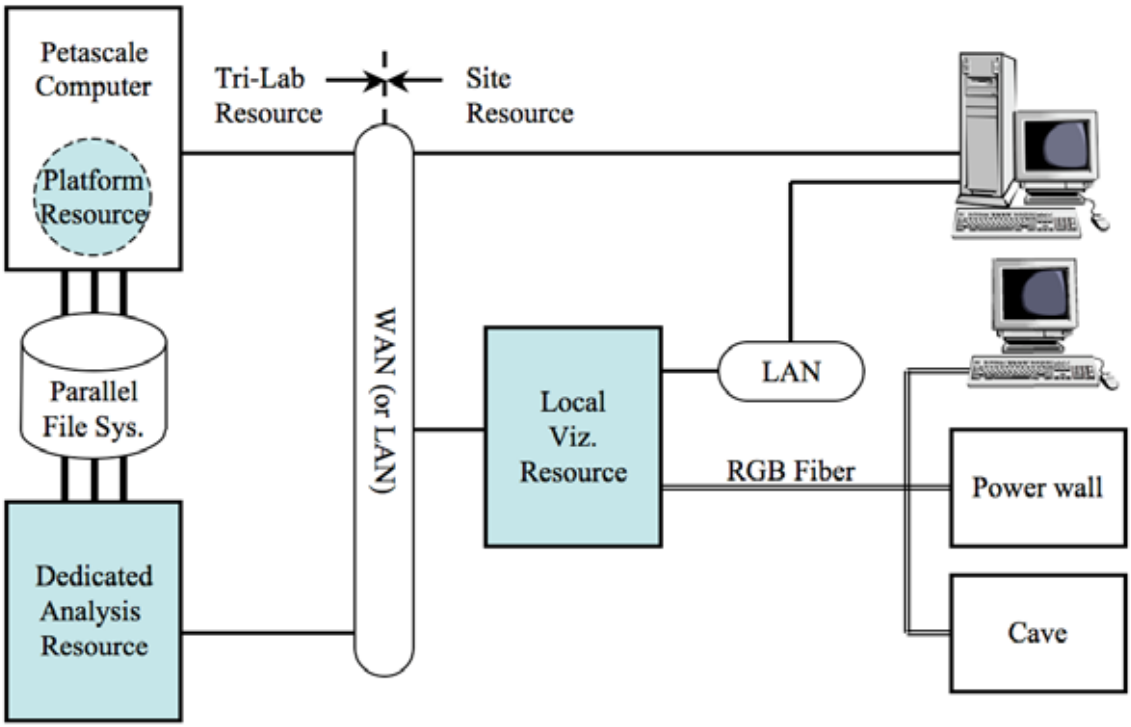
We envision a near future in which discovery is no longer limited by the speed of the disk. A future in which rich data is written to disk by a combination of simulation code and analysis code, and software tools allow more complex investigation of data required by ASC's—transformation, V&V, and other—efforts. We must enrich the ways we can interact with data as well. Flexible tools that promote investigation of sets of related runs, as well as comparison of many runs, will promote investigation and understanding of the data.

Customer needs and ASC's increased mission emphasis on V&V require us to be more active in adding value to the data that is written to disk. In addition to dumps of data at specific time steps, our customers are demanding better artifacts from the simulation for those time steps that are not written to disk. A combination of regular dumps of high-fidelity data, images, movies, computed data (such as isosurfaces), and tracked features is required to bring the full impact of a simulation's results. Without richer artifacts from large simulations, we are throwing away valuable data and squandering compute resources.

The data analysis environment at the Tri-labs includes dedicated visualization and rendering hardware, distributed memory visualization software (running on either visualization clusters or the platforms), desktop delivery to offices, and facilities for group interaction. (The figure below shows the preferred flow for large-data visualization.) ASC codes are utilized by a broad range of customers, so the data analysis environment must consist of tools and hardware that support an array of efforts.

At present, this environment successfully supports investigation of large data, but as the ASC program enables V&V of petascale results, customers will need tools that promote investigation and analysis of larger, more complex data, comparisons of multiple results, and increased analysis capabilities on a range of data. We believe that appropriate investment in hardware and

software tools will provide an effective petascale analysis environment when supported by appropriate storage, I/O, and networking solutions.



The current preferred data flow for visualizing large data is as follows: simulation data is read from a parallel file system into an analysis resource at the platform, which may be a dedicated analysis resource with shared access to the parallel file system or a platform analysis resource (in which part of the platform can be utilized for analysis jobs, including visualization). In either case, the key characteristics of such a resource are that (1) the resource supports interactive use cases, and (2) the resource has direct high-speed access to the data so that the data does not need to be moved. The data is then analyzed or visualized, which typically results in data—geometry, images, or other data—that is much smaller than the original data. Final rendering can take place on the remote resource or the local resource, depending upon the requirements of the final destination (desktop, powerwall, cave, or other facility).

Petascale data makes everything more difficult—from reading the data from disk to making sense of results. Investment in hardware will address, in part, the size of the data and will continue the success of solutions currently serving the community. For the mid and long term, investment in analysis and visualization software will address complexity and enable discoveries. Analysis techniques, such as those that enable V&V, will become increasingly important as part of petascale data analysis.

We note that the challenges of petascale data analysis are faced by a range of institutions. The needs of these institutions are not identical, as they are necessarily tied to applications, research domains, and resource constraints. Thus, an important strategy for achieving a petascale data analysis environment will be partnering with DOE Office of Science and other appropriate U.S. government agencies, industrial partners, and international collaborators (AWE, CEA) to maximize leveraging of existing R&D efforts. Complimentary goals on petascale data analysis are being pursued through SciDAC's Visualization and Analytics Center for Enabling Technologies (VACET) and SciDAC's Institute for Ultra-Scale Visualization, and we expect existing ties to these centers to enable effective partnership.

2 DEVELOPMENT/DEPLOYMENT AREAS

Successful petascale data analysis requires balanced investment in hardware and software. We address each of these areas individually, though in practice they are part of a unified environment providing service to our customers.

Petascale data analysis faces the following broad challenges, which we address in detail in later sections:

- **Challenges in visualizing and understanding a single petascale run.** ASC's petascale platforms will produce more complex data by enabling simulations to operate on higher resolution of meshes, higher dimensionality of data, and higher resolution in time. The complex interaction of when, where, and why phenomena occur can only be understood by enabling more interactive, intuitive, and insightful analysis of our data.

- **Challenges in comparing sets of closely related (ensembles) of runs.** At the heart of V&V lies the ability to compare sets of related data, so we can understand how changes in codes affect results. In particular, the ability to detect and understand how changes in codes or initial conditions produce different results is crucial to code validation, as well as to understanding physics. Comparing simulated results with sensor data or ideal solutions is essential to this process, as is detecting features and anomalies that are of interest. Automated and assisted techniques can make it possible to find areas that require further study by analysts, who will be unable to sift through petabytes of data—especially when considering more than one set of results.

- **Challenges posed by the architecture itself, particularly advanced architectures.** As discussed below, one option is for post-processing visualization and analysis is to be performed on the petascale platform (rather than on a separate, specialized visualization system). If this option is chosen, visualization software will need to be extended to run on multicore platforms. In addition, multi- and many-core architectures pose the memory wall problem: although the number of processors increases, total bandwidth is not, so current bandwidth-constrained algorithms will not be able to use the increased processing power. Many-core architectures may provide the opportunity to implement visualization and analysis algorithms that has been impossible in the past due to the algorithms' computational complexity.

2.1 INVESTMENT IN HARDWARE

Appropriate investment in hardware is crucial to the success of a petascale data analysis environment. The existing Tri-lab strategy of a combination of shared resources at the platform and local resources at individual sites will support the community of Tri-lab petascale customers and should be continued. Shared sites can optimize for a community of Tri-lab users, while individual sites can provide a flexible environment for local users that satisfies a need for on demand, interactive resources. This strategy will make the most effective use of the investment in petascale computing by providing a range of resources that will optimize compute time as well as customer effectiveness.

We note that the platforms will have individual strategies for hardware acquisition based on platform architecture, budget, usage models, and customer needs. In particular, we expect that there will be cases in which the petascale platform itself is utilized for both in situ and post-processing of data, and there will be cases in which a specialized, co-located visualization platform will be available for post-processing. Budget and phasing may dictate that a petascale platform be used for analysis and visualization in the early stages of deployment, with separate visualization hardware to come online at a later phase. This is a complex issue that can only be tackled within a specific platform strategy document, and it will not be fully addressed here.

2.1.1 CONCERN: Shared Tri-Lab Visualization Resources Near Petascale Platforms Will Be Needed for Visualization of the Data Generated from Petascale Runs

Moving full scale petascale data will be enormously inefficient, so appropriate visualization and analysis resources must have sufficient access to data near the platform. Solutions such as shared parallel file systems are addressed in the *ASC Implementation Plan* or individual platform planning documents, such as the Request for Proposal (RFP).

2.1.1.1 STRATEGY: Include Appropriate Visualization and Analysis Hardware and Infrastructure Consistent with Platform Costs. Deploy and Upgrade Such Hardware As Needed.

This continues the successful strategy of providing post-processing resources at the platform. The historical rule of thumb has been 6% of a capability platform should be budgeted for visualization and analysis hardware. However, platforms at petascale and beyond may require a change in this estimate. We will have to determine the appropriate ratios for new architectures based on intelligent application of historical measures of performance (FLINS, FLOPS, etc.)

We strongly recommend that visualization and analysis hardware resources be consistent with platform costs. Visualization and analysis resources are often tacked on as an afterthought, especially when budgeting. Petascale demands that integrated solutions be designed and delivered in concert, so that petascale results can be analyzed. Note that this includes not only the platform-specific visualization and analysis resource and the rendering resource, but also the storage, I/O system, and the LANs (local area networks) and WANs connecting the compute platforms to the desktop or the visualization cluster. The RFP for Sequoia includes this recommendation, though the visualization resource has not yet been allocated. These will all need to be scaled to match the increase in the amount of data. Because each petascale platform is unique, platform-specific planning documents should be referenced for specific details outside the scope of this document.

Priority: Essential. There *must* be a visualization and analysis resource for the platform. Whether that resource is Option 1 or Option 2 will be platform or site specific. Difficulty and cost are shown below.

Option 1. Dedicate a portion of the compute platform to visualization

Priority: (see above) **Difficulty: Medium** **Cost: $$$**

One current solution is to dedicate some portion of the nodes on the compute platform to visualization and analysis. However, the new petascale capability platforms are non-standard architecture, and if we want to fully exploit their capability, it will be necessary to port the production visualization software to the multi-/many-core or hybrid architecture, and this may be difficult (this is discussed further below). It will be possible on some architectures, such as that of

Roadrunner, to use part of the base system (which is a conventional cluster) for visualization without significant modification of visualization and analysis tools. In addition, usage models for petascale platforms would have to accommodate the interactive utilization model of visualization jobs, a much different usage model from computation. The opportunity cost of using the petascale platform in this way may be acceptable, if this is the only practical way to provide visualization and analysis resources at the platform. It is likely that the rollouts of computation and visualization hardware will not align, and it may be necessary to provide this type of service as an interim solution.

Option 2. Stand up a dedicated visualization resource near the compute platform

Priority: (see above) **Difficulty: Easy** **Cost: $$$**

A second option is to set up an appropriately sized conventional visualization cluster sharing a parallel file system with the compute platform. This provides a flexible platform—it can serve many use cases, run standard software, and can be upgraded at a rate different than that of the petascale platform. Such clusters have the advantage of incorporating graphics hardware, to increase performance. In addition, it is a solution that we know how to implement.

2.1.2 CONCERN: Sufficient Infrastructure Will Be Needed to Support Remote Access and to Support Access to the File System and Other Local Compute Resources

Users who are not at the site where the petascale platform is located need reliable and effective remote access to their visualized data. The current method for this is via the ASC secure WAN. We expect to see an increase in geometry size with the increase in data size, and this particular use case will put more demand on the resource. In the file systems area, I/O rates and latency from disk are improving, although not as fast as compute power, and the number of lines in to the clusters can be adjusted upwards.

Concerns touching on the secure WAN at petascale are detailed in the Networks and Interconnects section of this document. Concerns touching on I/O and file systems at petascale are detailed in the I/O section of this document.

2.1.2.1 STRATEGY: Define and Impose Requirements on Networking, File Systems, and Storage Areas

Priority: Essential **Difficulty: Easy** **Cost: $**

Networking services, I/O bandwidth improvements, and issues mentioned in the File Systems and Storage sections of this document will have a direct impact on visualization and analysis capabilities. We note this concern here to emphasize the interdependence of visualization and analysis with the performance of other areas. Reference platform-specific planning documents for system specifics and relative difficulty.

2.1.3 CONCERN: Local Rendering and Visualization Hardware Is Needed at Each Site

A range of simulation sizes must be supported in a petascale computation environment. Not all runs are at petascale. Often a petascale run is preceded by smaller studies that provide input for the petascale runs that increases the effectiveness and efficiency of the petascale platform. Thus, sites require appropriate on-site hardware resources to service high-use-case customers for large data.

Also, the data flow model in use for some facilities calls for local rendering after the data has been processed on a remote platform (see data flow figure in Introduction and Background section). In general, local resources will support on demand, interactive work on local data and data coming from capacity platforms, as well as assisting investigation of data residing on remote resources.

2.1.3.1 STRATEGY: Include appropriate local rendering and capability visualization and analysis resources in site planning. Coordinate as needed across the sites.

Priority: Essential **Difficulty: Easy** **Cost: $–$$$**

We strongly recommend appropriately sized local hardware resources, including the rendering and capacity visualization clusters and desktops, the networking and I/O resources that enable remote access of shared Tri-lab computing resources, appropriate connectivity to the analysts' desktops, and the visualization facilities used to view the results of visualization and analysis. These should be coordinated with other local resources and customers. This addresses the users' need for a high-availability, dedicated resource for data analysis and visualization with shared access to the data, and can be used for local rendering as well as local visualization of capacity runs. Note that cost will depend upon specific site needs. Due to the individual nature of the work performed by user communities at each lab, these facilities should be provided per local requirements.

2.1.4 CONCERN: The Visualization and Rendering Platforms, with Current Technology, May Not Have Sufficient Capability to Effectively Deal with Petascale Data and Geometry

The increasing size of the data will be a bandwidth challenge into the visualization and rendering platforms over the timeline of this document. This will impact the speed and interactivity of the visualization/rendering process and will impact the effectiveness of any analysis performed on petascale data. In addition, we expect that equipment in use during Phase I could benefit greatly from improvements in hardware that occur in the short term.

2.1.4.1 STRATEGY: Upgrade Performance of Local Hardware over Time, Consistent with Platform Capabilities and Applications Requirements

Priority: Highly Important **Difficulty: Easy** **Cost: $-$$$**

This is particularly critical in any area affecting bandwidth into the processors, such as the secure WAN, I/O or PCIe into the graphics processors. Because visualization and rendering are already impacted by insufficient bandwidth, this problem will only worsen with petascale, and any improvement in these technologies will help the situation and should be exploited.

Existing systems and those purchased during Phase I of this document's timeline should be considered for upgrade as petascale needs increase. Hardware components such as PCIe interfaces, which will double in speed twice over the next five years, may provide easy solutions to bandwidth problems encountered as users manipulate more and more petascale data. Graphics cards have been doubling in speed every year and may provide the ability to render faster and may also provide other means of analyzing user data. These solutions could potentially provide improved data analysis and visualization services to end users and should be considered where appropriate for customers. An analogous situation on a platform would be the upgrade to faster processors to provide more compute power.

2.2 CONTINUED INVESTMENT IN ADVANCED ANALYSIS SOFTWARE

With appropriate investment in hardware, as outlined in the previous section, we will be able to deliver images and data to analysts. To promote understanding of that data, we need additional investment in the software tools that can help investigate and understand Petascale data. We believe that, in general, current distributed memory algorithms and visualization software will scale well on petascale-sized hardware, but it will be increasingly difficult to understand and explore the vast amounts of data in the relatively ad hoc way it is done today. It makes sense to harness the power of computation to assist analysis of the vast stores of information that petascale computation will produce. This will help realize the enormous potential of ASC investments and will allow customers to spend time thinking, rather than wrestling with large data.

We note that V&V/QMU analysis presents unique challenges for extreme data sizes. For example, how does a scientist understand and quantify the changes that result when new algorithms or physics are introduced to an existing code? How does one understand the qualitative differences between different codes? How does one understand quantitative differences between solutions in a parameter space? These questions require a combination of qualitative and quantitative analysis tools to answer them.

2.2.1 CONCERN: Data Output from the Simulation Will Be Limited Due to the I/O Bottleneck

Currently, due to the disparity between disk I/O speed and compute speed, simulations do not retain high-fidelity data, because it is not possible to write out the breadth of information that is computed. This problem will be exacerbated as we move to petascale, and simply throwing away computed data is not an acceptable solution. We need to retain as much information as possible from the Petascale runs, so that further study of the information yields as much knowledge as possible, whether results are viewed in isolation or in comparison to other results.

As noted before, I/O speeds will continue to be a bottleneck for the analysis pipeline. Distributed memory algorithms and rendering scale well but are constrained by how fast we can get information off of disks. We shall rely on other investment areas to drive investment in I/O, but we expect that even with appropriate advancement in I/O, data at petascale and beyond require fundamentally new approaches to analysis and visualization. We must develop techniques that intelligently process extremely large results, and allow comparison of large numbers of such results. In short, the file system will be an inadequate method for communicating data through the analysis pipeline. We see this already, as analysts currently write out very sparse data—even at data below petascale. If we are to enable true characterization, feature detection and analysis of our petascale results, we must fundamentally alter this method of working. Intelligent sampling of the data will yield far more useful and meaningful results than the sparse sampling that currently goes on.

If we are to support investigation, comparison, and quantitative analysis (including V&V) on petascale data, it will be necessary to write out rich information from these large runs. If we do not, much of the value of the computation will be unavailable for further study, thus devaluing the resource that most constrains us—compute time.

2.2.1.1 STRATEGY: Analyze the Data while It Is Still on the System, before Writing to Disk. This Will Reduce the Output while Retaining the Information.

Priority: Essential **Difficulty: Medium** **Cost: $$** **Phase: II**

One option is in-situ analysis, in which some analysis occurs in partnership with a simulation, as it is running. In a perfect world, we would run extremely large simulations on high-fidelity models and preserve both high-fidelity information at regular intervals and additional features that are of interest. If sufficient analysis can be done on the data during the simulation, many important, high-fidelity features could be tracked and preserved in ways that are not possible at present. Effectively, this means doing away with the file system as the method of transferring data from the simulation to analysis, visualization, and post-processing tools. We will lose valuable results and our science will suffer if we rely on the file system in this arena. We must develop technical approaches that allow analysis and visualization tools to operate on high-fidelity data—much of which may not be written out. The better the data that the tools operate on, the better the feature detection and assisted discovery will be. This strategy requires partnering with code groups to define requirements and for implementation, and will it necessitate a higher degree of development cooperation than in the past.

2.2.1.2 STRATEGY: On-the-Fly Compression Techniques

Priority: Highly Important **Difficulty: Easy** **Cost: $**

We need to utilize compression technology to reduce the amount of data being written to files. This will include both lossy and lossless techniques, depending on the use case. By compressing the data, not only is less data stored on disk, but the effective I/O rates are improved because less data is being written and read from disk.

2.2.1.3 STRATEGY: Reorganization and Annotation of Data for Interactive Post-Processing

Priority: Highly Important **Difficulty: Easy** **Cost: $**

We need to organize and annotate the data so it can be read as quickly as possible. This includes organizing the data so that all the data that is necessary can be read in one large chunk, the data is annotated so that minimal amount of data is read to perform the operation, and the data is organized so that approximate representations of the data can be quickly shown to the user for interactive processing.

2.2.2 CONCERN: Petascale Data Will Be Too Large to Effectively Explore

Assuming that rich simulation data can be written to disk, effectively exploring that data presents its own challenge. Like the Internet, petascale data will become useful when we can assist discovery with powerful tools that utilize the power of the visualization and analysis resources. Trolling through these vast stores of information by hand will not scale.

In addition, the complexity of the results requires tools to understand the complex relationships within the data. Certainly, a human can understand how varying a single variable can influence results, but optimization, V&V, and complex interplay between multi-physics codes can only be understood through harnessing visualization and analysis. This problem increases when we address the customer need to compare several results, or when we address the V&V need to investigate ensembles of runs. We need computational help to determine areas of interest within the data and to assist in exploring, analyzing, and annotating the data. Techniques viable at terascale will not

suffice, simply because the quantity of data will outpace the ability of infrastructure to access the data.

At the heart of V&V lies the ability to compare sets of related data so we can understand how changes in codes affect results. In particular, the ability to detect and understand how changes in codes or initial conditions produce different results is crucial to code validation as well as to understanding physics. Comparing simulated results with sensor data or ideal solutions is essential to this process, as is detecting features and anomalies that are of interest. Automated and assisted techniques can make it possible to find areas that require further study by analysts, who will be unable to sift through petabytes of data—especially when considering more than one set of results.

2.2.2.1 STRATEGY: Invest in Software Tools that promote Comparative Analysis and Feature Extraction

Priority: Highly Important **Difficulty: Medium** **Cost: $**

Simulation results must not be analyzed in isolation, so we must develop better methods of comparative analysis. Quantitative and qualitative comparison of different results is crucial to understanding the nature of differences between results. Assisted discovery of trends and features in the data will make it possible to understand large and small scale areas of interest between results.

Finally, as noted before, assisted discovery of features within data is a crucial component to understanding results. Assuming that we can achieve the goal of providing good input to feature extraction technologies (in-situ analysis, for example), there remain significant challenges in extracting features from petascale-sized data. Being able to 'Google your data' for interesting features, or having software that automatically identifies potential interesting features, is crucial to helping analysts investigate and understand the data these platforms produce.

Enabling comparison of petascale simulation results with exact solutions that can be calculated by the post-processing tools, for example, promotes V&V without requiring an exact solution data set to reside on disk.

2.2.2.2 STRATEGY: Invest in Software Tools that Promote Investigation of Ensembles of Runs

Priority: Highly Important **Difficulty: Medium** **Cost: $**

Investigating 'ensembles of runs', which is crucial to V&V/QMU and understanding results in general, presents unique data access, assisted discovery, and I/O requirements. Investigating a single petascale run is difficult; investigating hundreds of runs presents its own unique challenges.

2.2.3 CONCERN: Advanced Hardware Platform Requires Hardware-Specific Software Solutions

The type of computer architectures that will support ASC simulation and modeling capabilities at petascale are undergoing a revolutionary transition. The next generation of processor architectures will continue to increase performance but will do so in a disruptive way that will require changes to existing visualization applications at the algorithmic and coding levels, just as changes will be required for ASC simulation codes. These platforms are even more I/O bound than current ones, as they are expected to perform orders of magnitude faster than the I/O pipelines.

2.2.3.1 STRATEGY: Use the Advanced Platform for Visualization and Analysis

Priority: Highly Important **Difficulty: Medium** **Cost: $**

We need to use the petascale platform for visualization and analysis. Specifically, we need to dedicate a portion of the cycles on the platform to this task. We suggest allocating both a small percentage of the usage of the platform to this task as well as dedicating a portion of nodes to visualization and analysis. We need to integrate portions of visualization and analysis software into simulation runs. Specifically, techniques including data analysis and feature extraction algorithms that were described previously are critical to visualizing petascale results on these architectures.

2.2.4 CONCERN: Commodity Hardware Technology Continuously Advances and May Provide Opportunities for Transformational Capability in Visualization and Analysis

There are several technologies emerging or expected to improve greatly over the next ten years. Among these trends are the improvement in GPU performance, the expected collocation of the GPU and CPU onto a single die on some chipsets, and expected improvements in the Cell and similar processors. All of these have the potential to improve visualization and rendering performance on ASC problems.

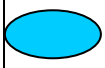
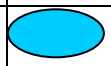
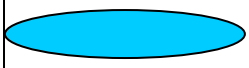
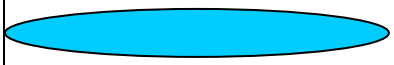
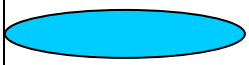
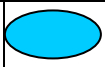
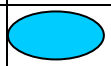
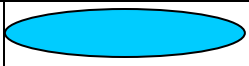
Over the last five years, GPUs have been improving in processing power at a rate of 2x to 2.5x per year. This means there will be impressive compute power available at the end of the viz pipeline, disproportionate to the rest of the system. Finding ways to exploit this power to directly impact our users may provide breakthrough capabilities that enable better understanding of ASC data. A second advance expected in the next few years is the incorporation of GPUs onto the same die as the CPU. The collocation of CPU and GPU on the same die presents an opportunity. The bandwidth will be much greater coming from the CPU into the GPU, thus addressing the PCIe/GPU imbalance. Streaming visualization/rendering techniques could be used to exploit this. A third area that must be kept in view is Cell and similar accelerator technology as it advances. Other areas of interest may present themselves in later years.

2.2.4.1 STRATEGY: Continually Stay Abreast of Commodity Hardware Developments and Investigate Ways In Which Software and Algorithms Designed for Such Hardware Could Produce Benefits that Could Be Applied to ASC Petascale Needs

Priority: Highly Important **Difficulty: Hard** **Cost: $**

Because these developments have the potential to provide solutions to the needs of our large data customers, we must devote resources to staying current with advances in hardware. By collaborating with colleagues, staying informed, and experimenting as necessary with new hardware, there is an opportunity to develop software that takes advantage of specific capabilities of leading edge hardware in novel ways to solve important problems for our users. We expect this to be a user-centered approach,in which knowledge of the state-of-the-art combined with intimate understanding of user requirements will allow us to make informed decisions about applying these technologies.

3 TIMELINE SUMMARY

Strategy	Activity	Near Term	Medium Term	Long Term	Comments
2.1.1.1	Include appropriate visualization and analysis hardware and infrastructure in consistent with platform costs. Deploy and upgrade such hardware as needed.				
2.1.2.1	Define and impose requirements on networking, file systems, and storage areas				
2.1.3.1	Include appropriate local rendering and capability visualization and analysis resources in site planning. Coordinate as needed across the sites.				
2.1.4.1	Upgrade performance of local hardware over time, consistent with platform capabilities and applications requirements.				
2.2.1.1	Analyze the data while it is still on the system, before writing to disk. This will reduce the output while retaining the information.				
2.2.1.2	On-the-fly compression techniques				
2.2.1.3	Reorganization and annotation of data for interactive post-processing				
2.2.2.1	Invest in software tools that promote comparative analysis and feature extraction				
2.2.2.2	Invest in software tools that promote investigation of ensembles of runs				
2.2.3.1	Use the advanced platform for visualization and analysis				
2.2.4.1	Continually stay abreast of commodity hardware developments				

I/O, FILE SYSTEMS, AND STORAGE

1 INTRODUCTION AND BACKGROUND

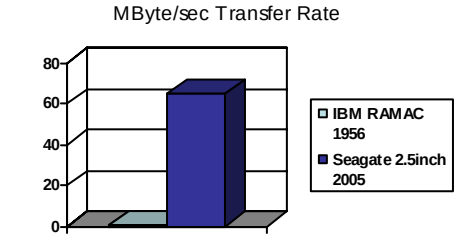
To understand the storage, file systems, and I/O area, it is important to understand our current position (from which we gain much leverage). It is also important to understand some basic trends in storage and HPC demands on I/O. Additionally, with today's declining budgets, it is important to partner with other HPC entities to provide as many solutions as possible to spread the cost burden.

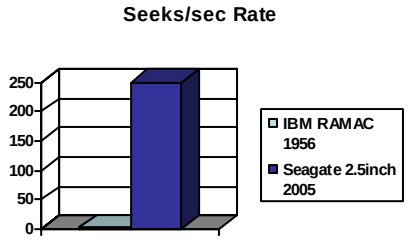
Past Leverage

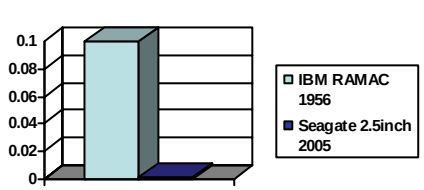
HPSS has become a dependable and high-performance archive service for our environments. We have moved from an era of few proprietary scalable file systems to multiple competitive sources for open systems based scalable file systems, with globally shared scalable file systems deployed. All of the ASC supported solutions, HPSS, PanFS, Lustre, and GPFS, are currently in production with the ASC computing environment and are enabling successful use of the computing environments at the current few hundred teraflop scale. Bandwidth scaling for large I/O operations has been achieved and work is under way to scale other aspects of file system solutions. Important and innovative R&D at university alliances is paying off by providing: (1) metadata and security scaling research, (2) NFS version 4 (NFSv4) and parallel NFS (pNFS) features for our use, (3) solutions for small and very noncontiguous I/O, and (4) research into the feasibility of leveraging global parallel file system technology for archiving.

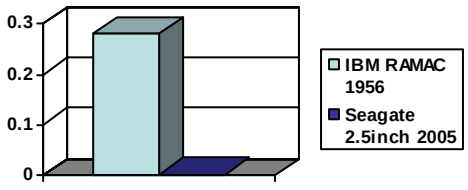
Storage Trends

It is well known that individual storage devices are getting denser at an amazing rate, keeping up with the ever faster speeds of processors. It is also well known that bandwidth to and from individual storage devices is getting faster, but at an alarmingly slower rate than the increase in density of the devices. Additionally, latencies to storage devices, both disk and tape, are not getting faster at anywhere near the pace represented by Moore's law. In the following graphs, you can see that both CPU speeds and density of disk drives have gone up by between four and five orders of magnitude in the last 50 years.









Unfortunately, the data transfer rates and seek rates of disk drives have only gone up by two orders of magnitude. This is one of the fundamentals that underlie the difficulties that exist in the I/O area. Scaling up of I/O performance has lagged behind the meteoric rise in CPU performance. Also, the economics of storage device use in the broad market is not significantly driving up individual device reliability and availability. Multi-disk subsystems are becoming more reliable and available, but not at the pace dictated by Moore's law.

We now see supercomputers with tens to hundreds of thousands of processors and anticipate machines with millions of process elements deployed in our environments in the near future. The possibility of billion-way parallelism seems likely. The shear scale of components creates many problems for the I/O area. It means that to get efficient writes to storage, data will need to be gathered from many more processing elements. With this trend, this problem is likely to get even worse. Machines with millions of uniform processors, each with hundreds of relatively slow cores, or machines with a few thousand nodes, each node capable of tens of teraflops, are likely future platforms. High-performance I/O requirements present major scaling issues on today's systems and will, in the future, present scaling issues well beyond today's current set of issues. The archives will be faced with managing exabytes of data and difficult component scaling issues as well. To assist in data and productivity management, higher levels of integration, new access methods, and innovative solutions for dealing with widely varying workloads are needed.

A New Era of Leverage (Collaborative Sources for Addressing Our Issues)
In today's budget climate, it is necessary to leverage and collaborate with other funders of technology solutions. In the area of file systems and I/O R&D, we can leverage, the Tri-lab's Path Forward for file systems (Lustre), our existing good influence in file systems and storage industry, Tri-lab's university investments, and Sandia LWFS work. Additionally, the HEC FSIO organization which manages 28 file systems and I/O projects, the SciDAC Petascale Data Management Institute and Data Management Centers, the NSA ACS program, various multi-agency collaborations, and influence in standards bodies can be leveraged. Whenever possible, Tri-lab I/O projects will leverage the open source software community and projects. When Tri-lab development is determined to be necessary, every attempt will be made to release the resulting product as open source in order to benefit the entire HPC community. To find solutions, the Tri-lab project teams will ensure that all avenues of external funding/collaboration are considered.

Overall Status of I/O, File Systems, and Storage Area
In the I/O, file systems, and storage area, we have chosen to break down development/deployment efforts into three areas: file systems and I/O, archive, and broad sharing. The breakdown is based on function/application/use that drives the characteristics of the solutions and segregates the issues and concerns. File systems and I/O concentrates on providing global parallel file systems used by supercomputers and analysis systems for scratch and data analysis/data mining applications. It also addresses I/O libraries and other file I/O related issues and concerns. Archive concentrates on HPSS and related archive issues. The broad sharing category covers NFS/DFS and related services. Within these three areas, we have differentiated efforts as being short term (e.g., FY08–FY10), medium term (e.g., FY11–FY13), and longer term (e.g., FY14–FY16).

File Systems and I/O
At the beginning of the ASCI program, the file systems and I/O area was essentially nonexistent. The parallel supercomputing industry had not developed any general purpose parallel file systems prior to the ASCI program. There were some special purpose parallel I/O solutions, but these were

far from scalable and general purpose enough for the goals of the ASCI program. The ASCI program investments in parallel I/O and file systems were groundbreaking and have catalyzed the HPC industry into providing multiple competitive solutions that are scalable for many I/O workloads. These solutions are being used at thousands of sites worldwide and are successfully deployed and being used today in our multi-hundred teraflop environments. There are many concerns in file systems and I/O in the areas of small/unaligned I/O, metadata scaling, QoS, manageability, and RAS.

As was described above, the base building blocks, disk technology, that underlie file systems are not getting more capable in the desired performance metrics at the same pace as are processors and even memory. This has the effect of an ever-widening gap in performance. Due to the past rapid increase in processor clocks and the current rapid increase in processor cores, the base building blocks for supercomputers are aiding in our ability to build faster compute platforms, but the base building blocks for file systems have not gotten appreciably faster for a decade. The effect is that parallel file systems must get wider in parallelism at a faster rate than processors in our supercomputers. This scale-out trend of file systems implies that vigilance in all concern areas affected by scale is prudent. Our successful efforts to make file systems scale for some workloads has allowed the HPC world to get productive work from our current compute environments, but until either a fundamental breakthrough in storage technology is made or the industry trends for the base building blocks for file systems changes fundamentally, file systems will be a concern in areas affected by scale. File systems also suffer from the normal issues in that industry solutions are focused on sweet spot sized installations, and the ASC environments are somewhat larger and remain at the edge of what industry is willing to provide.

There has been a lot of effort to not only successfully use parallel file systems in production at the hundreds of teraflops scale and scalable performance for some workloads, but also to manage the gaps and prime R&D in all the concern areas of file systems. In the File Systems and I/O section below, for the reasons above, you will see that there are many concerns, but due to the diligent management of gaps in this area, few of these concerns have immediate dire consequences for early petascale systems. There are some good strategies to keep these ever-widening gaps under control and get us well into petascale computing (and even approaching exascale).

Archive

The Tri-labs have been doing world class archiving for decades. Even parallel archiving was being done before the ASCI program was established. As was stated before, the investments in HPSS to provide our archiving technological needs allows our HPSS systems to provide world-class archives to our ASC environments today and for the last decade. Our archives are of the largest and most capable in the world and dwarf in usage and size those of all other non-intelligence agency scientific HPC sites. Additionally, unlike in file systems that use disk as their underlying technology, the underlying technology for archives, serpentine tape technology, has much closer to linear bandwidth gains as density increases. HPSS continues to evolve to meet our needs. The most pressing concerns in the areas of small file management and scalable metadata operations for the future of the archive service are well understood and planning has been done to address those concerns. In the Archive section below, we describe the top concerns for taking our archiving services into the multi-petaflop scale environment.

Broad File Sharing
In the area of broad file sharing, we do not need capabilities quite so far beyond what many much smaller computing sites need, thus making the solutions in this area far more off-the-shelf and making the concerns in this area far more deployment related and less R&D related. In the Broad File Sharing section below, two testing/deployment concerns are addressed.

2 DEVELOPMENT/DEPLOYMENT AREAS

2.1 FILE SYSTEMS AND I/O

As was mentioned above, many of the issues related to providing global parallel file systems concern the slow pace at which disk storage devices get faster and more agile compared to processing technology advancement. There are multiple I/O patterns in simulation activities that currently dominate the file system use. The primary I/O patterns are:

- N to 1 - N process writing/reading to one shared file.
- N to M - N processes writing/reading to M shared files where M is much less than N.
- N to N - N processes writing/reading to N files all into the same directory.

Additionally, our parallel file systems show other patterns similar to normal non-parallel file systems, like small file creates by one process by multiple users (like compiles, etc.), file tree walks, lots of file lists, creating tar files (doubling storage required), etc.

In addition to the current write-intensive, parallel checkpoint/defensive I/O workloads that have enormous scale and a mix of small and large I/Os, it is also likely that usage patterns for our machines may change, thereby changing the needs of the file system. The ASC program began with few applications that were scalable and only a small set of users. In future, simulation will be simple enough such that many users will use the machines and data analysis capabilities. Tools for easily manipulating very high dimension data at multi-terabytes in size will be needed (and indeed are lacking today). The productivity in dealing with, exploring, and learning about data must be raised in order to support the growing use of simulation.

In general, continued support of problem determination, working with vendors on fixes, regression testing, new version deployment, and other related activities will have to continue to keep this portion of the infrastructure healthy. This document assumes this level of support for tactical problems and planned growth in infrastructures. These are ongoing necessary costs for production environments. This document only calls out specific gap areas where investments beyond ongoing support costs are required.

2.1.1 CONCERN: Small Unaligned I/O Performance

The trend towards more and smaller memory processes mated with disk systems requiring larger blocks for efficiency combined with the high metadata workloads (which also drive small I/O requests) both imply that small and unaligned I/O is a serious issue. Past efforts, such as putting aggregation in I/O middleware, are not working well now and are not keeping pace with the dramatic rise in parallelism (multicore, etc.) and the smaller memory per process on newer advanced architectures.

2.1.1.1 STRATEGY: Multi-Agency HPC Scalable I/O Forwarding Layer/Async Offload

Priority: Essential **Difficulty: Medium** **Cost: $ (very highly shared with other agencies)**

One approach to the small unaligned I/O concern is to start a joint DOE Office of Science/ASC and other HPC site effort for common scalability I/O operation forwarding layer. This effort would provide aggregation and complete hardware asynchronous operation at the file system level while enabling the use of *any* parallel file system as the underlying storage mechanism. Office of Science labs face this concern and are willing to help. Additionally, the proposed work needs to be expanded through the addition of non-volatile storage to further enable asynchronous behavior. This solution could drastically reduce the peak I/O rates required of the file system and could aggregate both metadata and small unaligned I/O operations, thus helping in a very large way in addressing the small and unaligned I/O issue. This solution will make our extreme high-end computers look more like industry sweet spot capacity systems to the file system, which allows us to more easily utilize industry solutions. This work needs to be started as soon as possible. A DOE SC FASTOS proposal has been submitted. (The proposal is available on request.)

It is important to note that this strategy is highly leveraged for all the community partners that have already indicated their willingness to help. It also addresses multiple concerns, including this small unaligned I/O concern and metadata scaling. This effort could also be a perfect vehicle to fundamentally change how parallel applications access file systems, which could be the catalyst for a related follow-on effort—the proposed technology promotion strategy (see technology promotion effort below). These two efforts are the basis for a real strategy for staying ahead of the ever-widening performance gap between processor and disk technologies. As in all projects looking for an exit strategy to minimize long-term expensive people mortgages for support is advisable. This project will define the next generation of access methods for parallel file systems, and that new paradigm will be the starting point for the proposed technology promotion effort as described below that could provide the desired exit strategy for the I/O forwarding effort.

2.1.1.2 STRATEGY: Multi-Agency HPC File System Technology Promotion

Priority: Highly Important **Difficulty: Medium** **Cost: $$ (very highly shared with other agencies)**

A longer term follow-on project to the I/O forwarding project explained above involves the formation of a multi-agency file system technology promotion effort. One of the values of the ASCI PathForward program was the secondary effect such that if we pushed technology into one product, other vendors followed suit to keep up in the technology offering race. This has been very useful in the file systems area and has enabled multiple solutions for global parallel file system solutions for HPC sites. The government should consider re-enabling this technology pushing capability. To make such a technology promoting a reality, whatever vehicle chosen must be a credible threat to the global parallel file system vendors. The Office of Science is highly motivated to make PVFS a credible threat and enable it as a real alternative for their flagship sites, so PVFS may be a good candidate for this activity. The Office of Science, NSF, and DOD have all expressed interest in this concept. PVFS is also the most popular research parallel file system due to its deployment in user space and its simple design.

Given that the I/O forwarding project, explained above, will define a new scalable way for our applications/machines to talk to parallel file systems, it makes sense to use this as the basis for the

technology promotion project. PVFS is very well suited for this new access paradigm given that it was designed to disregard portions of the POSIX I/O interface that hurt scalability. This design is well mated to the work that will occur in redefining how parallel file systems are addressed by machines/applications. Additionally, if this technology is successfully promoted, we will drag along the commercial file system vendors to support this new paradigm and cause the industry to leap forward. The successful promotion effort is the exit strategy for supporting the I/O forwarding layer software for an extended time, which gives us a way out of having an expensive support mortgage.

2.1.1.3 STRATEGY: General Promotion of Use of Emerging NV Storage Technologies

Priority: Highly Important **Difficulty: Low** **Cost: $$**

With the cost of non-volatile memory dropping rapidly, Tri-lab architects need to investigate strategies for using this to our advantage. As mentioned above bringing NV storage to the I/O forwarding project is important. Additionally, inserting another layer of memory into the I/O hierarchy is an important idea that is mentioned above in the technology promotion area. This strategy extends the possible uses of NV storage beyond that mentioned in the I/O forwarding and technology promotion strategies to areas such as RAID disk controllers and on disks themselves. Options need to be studied with memory vendors, disk controller vendors, and system architects.

This solution is more of a risk mitigation and opportunity to reduce the effects of small unaligned I/O. It should be pursued no matter what machine, as it is trying to exploit commodity technologies to help all I/O solutions. Working on solutions with Flash now targeted at phase change memory in the 2011 time frame is probably the best approach, although there may be shorter term wins if the Flash write penalty can be managed or overcome.

2.1.2 CONCERN: Metadata Scaling and Extensibility

Scaling of metadata operations in both file systems and HPSS has not been solved. While some engineering solutions have been put forth for scaling some metadata operations including name space division, directory hashing, and directory splitting, there are still no solutions that address this area fundamentally. Given the number of processing elements of future supercomputers, this is one of the top problems in our future. Additionally, extensible metadata and alternative to tree-based organization and access for files needs to be explored to help with management of the billions of files we will have to manage. There has been basically no progress in this alternative tree-based metadata area.

2.1.2.1 STRATEGY: Creation of Parallel Metadata Related Tools

Priority: Important **Difficulty: Low** **Cost: $**

One tactical solution involves parallel metadata tools. Most metadata utilities are serial like “ls” and “find.” Parallel metadata utilities will need to be developed, perhaps by students, and users will need to be educated on how to use such utilities at least until the POSIX I/O API is enhanced to enable these parallel utilities naturally.

2.1.2.2 STRATEGY: Work with Existing File Systems Vendors for More Scalable Metadata

Priority: Highly Important **Difficult: Low** **Cost: $ (but success will depend on vendors)**

One tactical solution involves working with our file system vendors to enable more scalable metadata solutions in their file system products. Some file systems already offer first generation scalable metadata and have second generation designs done. The Tri-labs could work with current file system vendors to encourage more scalable metadata in future versions of their products.

There is not universal agreement that tactical investments here are the best course or about the lengths to which the Tri-labs should go in assisting current vendors to enhance current products due to stability, proprietary, and support concerns.

2.1.2.3 STRATEGY: Multi-Agency HPC Scalable I/O Forwarding Layer/Async Offload

Priority: Essential **Difficulty: Medium** **Cost: $ (very highly shared with other agencies)**

As was mentioned above in the small unaligned I/O concern area, the multi-agency scalable I/O forwarding project could help with metadata operation scaling. As was mentioned before, this is a strategy that addresses multiple concerns and an excellent precursor to the proposed multi-agency technology promotion effort. See Section 2.1.1.1 above for a full explanation of this strategy.

2.1.2.4 STRATEGY: Multi-Agency HPC File System Technology Promotion

Priority: Highly Important **Difficulty: Medium** **Cost: $ (very highly shared with other agencies)**

As was mentioned above in the small unaligned I/O concern area, the multi-agency technology promotion effort could help with metadata operation scaling. As was mentioned before, this is a strategy that addresses multiple concerns and is an excellent follow-on to the proposed Multi-agency scalable I/O forwarding effort. Additionally, there are multiple HEC and ASC sponsored projects such as the ASC UCSC Ceph scalable metadata project, the HECURA SUNY serial b-tree project, and the LANL CMU/ANL billion file directory projects that could use this technology pushing vehicle as a tool to prototype these ideas in. See Section 2.1.1.2 above for a full explanation of this strategy.

2.1.3 CONCERN: Quality of Service

With mixed workloads, one workload can adversely affect another, thus drastically affecting deterministic behavior on shared global storage systems. Two labs are already deploying centralized (or enterprise) file system services. The ultimate goal of this is to attach many workstations (via pNFS), mid-range clusters, and even premier machines to this global service. Economy of scale, both in terms of management and capital costs, will save us significant money relative to the alternative of one dedicated file system, with it's attached storage, per machine. Data retention and data management is augmented as the file system is independent of any machine, so it lives beyond the machine life. Sharing is augmented because the simulation platform's file system is available at the visualization platform and workstations. This is happening now and it will likely become more prevalent and core to our storage strategies. As this is leveraged more and more, the service will see greater and greater contention. We can offset this by growing the service, but the economy aspect is only fully realized if we allow the service to be somewhat oversubscribed.

We must manage the competing requests from many different platforms or be forced to accept a situation in which an individual, pNFS attached, workstation, for instance, can cause relatively large delays on our expensive clusters. This is because file systems manage storage so as to optimize throughput and not response. A large job, running on a large machine will be more sensitive to variances in response from the file system whereas smaller jobs and workstations are relatively insensitive. We must have a way to tell the file system managing all this storage what response delays a particular request can tolerate. We must have a way to tell the file system that, although it might not be optimal for storage to service a request now, it's important to the application that initiated the request. That is what QoS does. It augments requests with the information that a scheduler requires to balance responsiveness with throughput. It augments request schedulers to make a decision based on more than maximizing storage throughput. For us, importantly, it allows us to bound (within reason) response latencies. End-to-end QoS solutions are needed but non-trivial to build or implement.

2.1.3.1 STRATEGY: Work with Existing File System Vendors to Enhance Products for QoS

Priority: Highly Important **Difficulty: Low** **Cost: $ (but success will depend on vendors)**

A tactical solution involves working with our file system vendors to add QoS features to their products through special non-standard extensions.

There is not universal agreement that tactical investments here are the best course or about the lengths to which the Tri-labs should go in assisting current vendors to enhance current products due to stability, proprietary, and support concerns.

2.1.3.2 STRATEGY: Create External to the File System Scheduling System for Checkpointing

Priority: Important **Difficulty: Low** **Cost: $ (but success will depend on vendors)**

Another tactical solution involves developing a simple-minded scheduling window for dumps from jobs running on supercomputers. This solution would require application developers to utilize the facility that would be created. This would be difficult to enforce and could provide some short-term benefit, but in this case we would likely be leading our applications off into the weeds, so this should also be done very carefully.

2.1.3.3 STRATEGY: Multi-Agency HPC File System Technology Promotion

Priority: Highly Important **Difficulty: Medium** **Cost: $ (very highly shared with other agencies)**

This is a strategy that addresses multiple concerns. As was mentioned above in the small unaligned I/O concern area, the multi-agency technology promotion effort could help with metadata operation scaling. Given this effort can redefine how we talk to parallel file systems, it is an optimal place to do QoS work because QoS is quite pervasive in its implementation for requests of the file system. Additionally, the technology promotion vehicle would be an excellent tool to encourage R&D ideas to be tried while leveraging QoS projects and efforts including ASC UCSC Ceph, HECURA UCSC, CMU and SUNY efforts, POSIX High End Extensions work as well as other HEC FSIO

managed R&D projects. All of these R&D efforts attack the QoS issue, and all of these ideas could be prototyped and eventually merged into a technology pushing vehicle that would eventually be usable by ASC and would pressure the other file system market layers to follow suit. See Section 2.1.1.2 above for a full explanation of this strategy.

2.1.4 CONCERN: Lack of an At-Scale Testbed Technology/Tools Availability

Computer science researchers have no access to testbeds of large enough size that can be used in a dedicated way (root access with ability to wipe out/reload OS, etc.) for long enough periods of time. ASC and other supercomputer machines are installed and expected to be doing science ten minutes after the installation. This is unfortunate for the computer scientists because they often get no chance at trying out new concepts on these machines before they are deployed, and once the machines are doing programmatic science, no destructive computer science can be done. This is shortsighted, is often driven largely by program or lab marketing, and is simply nonproductive from the computer science point of view. Facilities or simulation tools need to be provided.

2.1.4.1 STRATEGY: Creation of a Virtualized Test Framework to Simulate Scale

Priority: Highly Important	**Difficulty: Medium**	**Cost: $ (could be highly shared)**

A user space parallel test tool that can simulate 10 million-way parallelism on existing 10–100k way parallel machines could be developed, possibly using virtualization techniques. We should seek out partners on this as well, since NSF, Office of Science, or other agencies all have this same issue and concern. This has been a long-standing issue in the HEC FSIO organization.

2.1.4.2 STRATEGY: Creation of an At-Scale CS Testbed

Priority: Important	**Difficulty: Hard**	**Cost: $$ (could possibly be shared)**

An at-scale hardware and software testbed could be jointly funded by ASC, HPC vendors, other interested agencies or a combination of all of these entities. There is not complete agreement about how such a facility should be run (i.e., multi-agency versus ASC controlled).

2.1.5 CONCERN: Reliability, Availability, Serviceability

RAS is a growing problem. The astronomic growth in disk capacity with only modest disk data rate means that classical RAID approaches are giving extremely long and growing rebuild times. Plus 2 and other multiple dimensional RAID technologies help by protecting during rebuild, but these are not fundamental solutions and they hurt performance for small, distributed, and unaligned I/O workloads, a particularly difficult workload without the added cost of more exotic RAID. This is because more exotic RAID schemes require more data to be collected for efficient write operations, which is further exacerbated by disk blocks getting larger over time. Object RAID does address this rebuild problem fundamentally but is not a solution all by itself. RAID 10 may end up being important given the abundance of disk drive capacity and the performance advantage it has for both writing and rebuilding. Additionally, a looming problem with all rebuild mechanisms is the enormous amount of data that needs to be read to do a rebuild. The mean time (bytes) to read error for many disk drives is getting dangerously close to the size of the disk itself. This means that unrecoverable read errors during a rebuild are becoming a huge worry. Needless to say, the reliability at scale issue is important.

2.1.5.1 STRATEGY: Encourage Current File System Vendors to Add RAS Features

Priority: Highly Important **Difficulty: Low** **Cost: $ (but success will depend on vendors)**

A tactical solution involves working with our file system vendors to add RAS features to their products. Some vendors have already addressed reliability at scale in many fundamental ways, including end-to-end reliability mechanisms, scalable rebuild schemes, and encoding for unrecoverable read bit errors, so it is not unreasonable to expect that pushing our other vendors to follow might work.

There is not universal agreement that tactical investments here are the best course or about the lengths to which the Tri-labs should go in assisting current vendors to enhance current products due to stability, proprietary, and support concerns.

2.1.5.2 STRATEGY: Multi-Agency HPC File System Technology Promotion

Priority: Highly Important **Difficulty: Hard** **Cost: $$ (highly shared)**

This is a strategy that addresses multiple concerns. As was mentioned above in the small unaligned I/O concern area, the multi-agency technology promotion effort could help with metadata operation scaling. In the case of RAS, PVFS, or other technology chosen for this possible technology pushing vehicle would be an excellent vehicle to encourage R&D ideas to be tried, especially ideas from the ASC and HEC UCSC Ceph RAS project, the CMU problem analysis project, the Wisconsin correctness work, and other HEC FSIO managed R&D projects. All of these R&D efforts attack RAS issue, and all of these ideas could be prototyped and eventually merged into a technology pushing vehicle. See Section 2.1.1.2 above for a full explanation of this strategy.

2.1.6 Manageability

The ability to manage, debug, tune, and diagnose hundreds of thousands of storage devices and massively parallel networks is difficult and getting harder. This area has seen much work by industry but full solutions are not in the hands of Tri-labs system administrators yet.

2.1.6.1 TACTIC: Encourage Existing File System Vendors to Add Manageability Features

Priority: Highly Important **Difficulty: Low** **Cost: $ (but success will depend on vendors)**

One tactical solution involves urging our file system vendors to add manageability features to their products. Some vendors have already addressed manageability at scale in some ways, so it is not unreasonable to expect that pushing our other vendors to follow might work to some extent. There is not universal agreement that tactical investments here are the best course or about the lengths to which the Tri-labs should go in assisting current vendors to enhance current products due to stability, proprietary, and support concerns.

2.1.6.2 STRATEGY: Multi-Agency HPC File System Technology Promotion

Priority: Highly Important **Difficulty: Hard** **Cost: $$ (highly shared)**

This is a strategy that addresses multiple concerns. As was mentioned above in the small unaligned I/O concern area, the multi-agency technology promotion effort could help with metadata operation

scaling. In the case of manageability, PVFS or another technology chosen for this possible technology pushing vehicle would be an excellent vehicle to encourage R&D ideas to be tried, especially ideas from the CMU Self* project, the correctness work being performed at Wisconsin, the Clemson File Systems modeling, the automated problem analysis work at PSU, and other HEC FSIO managed R&D projects. All of these R&D efforts attack the manageability issue, and all of these ideas could be prototyped and eventually merged into a technology pushing vehicle. See Section 2.1.1.2 above for a full explanation of this strategy.

2.2 ARCHIVE

Over the past 15 years, HPSS has become a dependable and high-performance archive service for our environments. The archive service has grown into the multi-petabyte range and continues to grow on an aggressive curve. It is now routine to store nearly half a petabyte in a single month. The Tri-labs' investment in the HPSS planning, development, and testing keeps HPSS healthy and relevant for our needs. Additionally, incredible tape technologies with data rates far outstripping disk technology and enormous densities of disk and tape technologies are contributing to the success of the archive service. Unfortunately, the ability to generate information in our supercomputer environments is outstripping our budget for storage media in the archive. Additionally the ability to generate file metadata is also pressing hard on our archive system resources.

2.2.1 CONCERN: Small File Management

HPSS small file management on media is an issue given the large numbers of small files generated at our sites. The client aggregation technique currently used is helping some, but there is a real need to have aggregation at the media end as well. Small file performance and management need to be improved in HPSS for both user experience and for long-term migration and management activities.

2.2.1.1 TACTIC: Complete HPSS R7.1, Which Has Small File Aggregation on Tape

Priority: Highly Important **Difficulty: Medium** **Cost: $**

This solution involves both use of the HPSS small file bundling on archive client now, and use of on-tape small file aggregation that is currently in the HPSS development plan. This functionality, along with improvements to the HPSS core server database metadata engine, is due with HPSS R7.1.

2.2.1.2 TACTIC: Complete HPSS R8.1, Which Has More Metadata Scaling Features

Priority: Highly Important **Difficulty: Medium** **Cost: $**

HPSS metadata scaling issues are a result of the number of small files as well as the sheer growth and high utilization of capacity and capability platforms in our centers. The work planned for HPSS metadata/core server scaling, scheduled for HPSS R8.1, will enhance our archives' ability to scale in support of petascale compute resources.

2.2.2 CONCERN: Sustainability of Long-Term Archive Strategy

It is unclear given the declining ASC budgets and the ever increasing compute power to re-compute if we can afford to continue with the current level of archive strategy support. The reliance on internal expensive mortgages for staff for maintenance of HPSS may not be sustainable.

2.2.2.1 STRATEGY: Investigate Alternative Commercial Options for Archive

Priority: Highly Important **Difficulty: Medium** **Cost: $**

In the medium term, a move to a commercial product for archiving should be explored. At a minimum, some strategic planning with DOE Office of Science, DOD, and NSF is in order to attempt to lower overall government costs in this area.

2.2.3 CONCERN: User Need for Automated Intra-Site Archive Movement Tools

HPSS users desire an automated way to move files between archives at the multiple ASC sites. The users would like a way to just specify files or file trees to be moved or replicated and have it occur in a completely automated and managed way with retry capability.

2.2.3.1 TACTIC: Creation of Automated Movement Tool

Priority: Highly Important **Difficulty: Easy** **Cost: $**

This tactical solution involves work on file transfer agents to automate the archive movement between labs. Work already done on GridFTP with HPSS may be a leverage point.

2.3 BROAD FILE SHARING

In addition to archiving and global parallel file systems, there is also a need for more general access to file sharing. This service is currently being provided via NFSv3. In the past there was also a DFS component involved in Tri-lab sharing. In addition to providing sharing of files on supercomputers, to workstations, and throughout the Tri-labs, there is also a desire to reduce the number of custom file system clients that work with our global parallel file systems in our sites, so the NFS protocol family is also being looked to for solutions. Additionally, as supercomputer-on-a-chip solutions become more and more powerful, it is not unreasonable to believe in a teraflop sized workstation on many desks. Providing good performing access to our global parallel file systems from these future platforms is also part of this broad sharing category.

2.3.1 CONCERN: Cost of Custom File System Client Code

Because the labs want to have access to a global parallel file system from premier machines, mid-range capacity clusters, and even workstations, the cost of dealing with custom kernel-based file system client code across the hundreds of thousands of clients using a myriad of operating systems will be prohibitive.

2.3.1.1 TACTIC: Secure Common Global File System Client

Priority: Highly Important **Difficulty: Medium** **Cost: $**

To remove the cost of maintaining hundreds of file system clients from our file system vendors and to enable more competition in the global parallel file system market, pNFS is being developed by industry. Additionally, pNFS may be a good way to get high-performance clients for future teraflop sized user workstations in their offices. If these very large workstations come to pass, the reliance on centralized disk systems to provide data to these workstations fast enough to make them useful will be vital. pNFS based on the secure NFSv4 is a likely solution for this need.

The tactical solution involves testing and eventual deployment of NFSv4 and pNFS products. Many NFSv4 and pNFS products are expected in FY08 and should start to be stable in FY09. There will

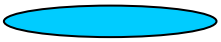
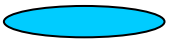
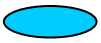
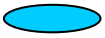
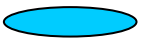
be opportunities to shape this solution in minor ways, at least for a small period of time, due to our ASC/HEC University of Michigan relationship. The University of Michigan is building and maintaining the pNFS over file and block capabilities for Linux. Additionally, we can leverage the Panasas efforts in the pNFS for OBSD T10 variant of pNFS they are building. Security portions of this project should be coordinated with the NNSA ICSI project, which is providing a common secure infrastructure for the entire weapons complex for uid/gid/principal and secure LDAP.

3 TIMELINE SUMMARY

As stated above, we have differentiated efforts as being short term (e.g., FY08–FY10), medium term (e.g., FY11–FY13), and longer term (e.g., FY14–FY16).

It is possible to provide delivery dates for items that are largely within our control, like HPSS releases and possibly even NFSv4 and pNFS releases from vendor information. For much of the file systems and I/O stack, the approach taken by the I/O community within the Tri-lab is to avoid generation of “home-grown” solutions where possible. Instead, it is to identify present and future problems and problem areas, then motivate researchers to deliver strategies and algorithms to mitigate or eliminate them. As part of that process, industry is involved in detail so as to be aware of the solutions and to guide the research so that it can be incorporated into products the Tri-lab might use.

This strategy is relatively new. For years (a decade or more), serious research in high-performance I/O has been absent. Many researchers have, frankly, fled the field. This was due to the combination of a lack of motivation from the user community (focus was on learning how to harness the compute capability), insufficient funding from the appropriate government agencies because of that, and the lack of a research vehicle causing any researcher to contemplate a significant investment in overhead to generate one. It’s only been within the last three years that the US government and the Tri-lab I/O community realized there were no longer ready solutions. It’s also only been within the last three years that real research vehicles have arrived; PVFS II, Lustre, and Ceph. The evolutionary research required to deploy the next generation of machines is only happening now. The community is beginning to reap the rewards of this. The security enhancements that are embedded in NFSv4 and the standards work on pNFS, for example. These examples, though, are only in a pre-production state. They are still being developed and adapted for large production use such as the Tri-lab requires. The work is being done, primarily, by others, and although motivated by the Tri-lab I/O community, it is not controlled by the Tri-lab I/O community. This is the price of working as a community and sharing the cost. For this reason, much of the timing information provided has little fidelity.

Strategy	Activity	Near Term	Medium Term	Long Term	Comments
2.1.1.1 2.1.2.3	Strategy: Multi-agency HPC scalable I/O forwarding layer/async offload				Strategy that addresses multiple concerns
2.1.1.2 2.1.2.4 2.1.3.3 2.1.5.2 2.1.6.2	Strategy: Multi-agency HPC file system technology promotion				Strategy that addresses many concerns, best if done as follow-on to I/O offload
2.1.1.3	Strategy: General promotion of use of emerging NV storage technologies				
2.1.2.1	Strategy: Creation of parallel metadata related tools				
2.1.2.2	Strategy: Work with existing file system vendors for more scalable metadata				
2.1.3.1	Strategy: Work with existing file system vendors to enhance products for QoS				
2.1.3.2	Strategy: Create external to the file system scheduling system for checkpointing				
2.1.4.1	Strategy: Creation of a virtualized test framework to simulate scale				
2.1.4.2	Strategy: Creation of an at-scale CS testbed				
2.1.5.1	Strategy: Encourage current file system vendors to add RAS features				
2.1.6.1	Tactic: Encourage existing file system vendors to add manageability features				
2.2.1.1	Tactic: Complete HPSS R7.1, which has small file aggregation on tape				
2.2.1.2	Tactic: Complete HPSS R8.1, which has more metadata scaling features				
2.2.2.1	Strategy: Investigate alternative commercial options for archive				
2.2.3.1	Tactic: Creation of automated movement tool				
2.3.1.1	Tactic: Secure common global file system client				

NETWORKS AND INTERCONNECTS

1 INTRODUCTION AND BACKGROUND

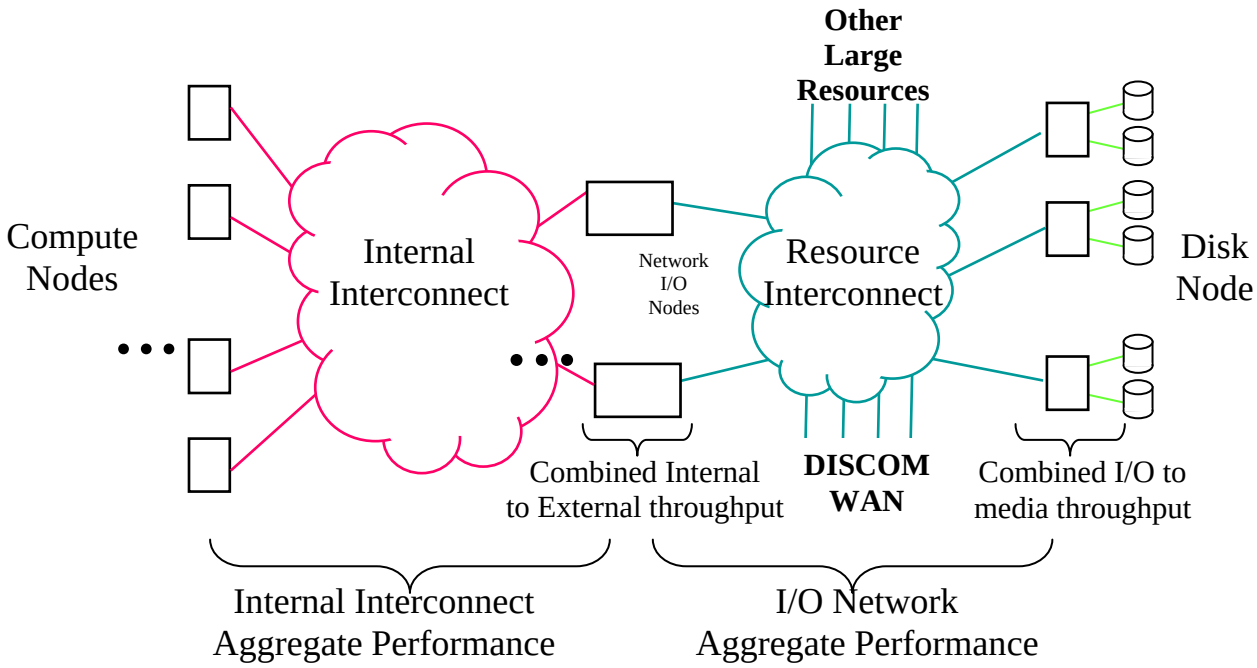
The network interconnect environment encompassed by the Tri-lab ASC network is a tightly coupled confederation of LANL, LLNL, and SNL classified supercomputing facilities. The environment scales in many dimensions. Each lab network environment spans from the central computing rooms to the user's desktops. The network performance starts at 1 gigabit/s at the desktops and increases to the multi-gigabyte/s interfaces in the core of the compute platforms. There are tens of thousands of interfaces in the compute core, hundreds of connections between large resources, thousands of desktops, and a small number of links in the WAN connections. Each of the local laboratories' networks is connected across secure 1100 mile, 10 gigabit/s wide area links.

2 DEVELOPMENT/DEPLOYMENT AREAS

In order to effectively address this wide variety of scales, this section is divided into three subsections. The Compute Interconnect subsection will consider the extremely high-performance networks that interconnect the compute nodes inside the largest platforms. The Resource Interconnect subsection will consider the network that interconnects the largest platforms with themselves, their resources such as storage, and the user's desktop. The WAN Interconnect subsection will consider the network that interconnects the three laboratories in a seamless environment for sharing remote resources. The figure below shows how these networks combine to provide the Tri-lab ASC compute environment.

2.1 WAN INTERCONNECT

The WAN interconnect between the ASC compute facilities consists of leased, private bandwidth and the associated equipment to securely interconnect the networks from each of the laboratories. The bandwidth currently consists of a 10-gigabit/s (Gbps) ring. Because the ASC mission is focused on classified products, the WAN links must be protected utilizing NSA approved Type 1 encrypted devices. The following four technical challenges represent the consensus thinking of the Tri-lab networking community; however, there are other challenges to designing a Tri-lab network that are not technical in nature but will have an impact on planning and implementation. These challenges are an uncertain budget future, the geographical disposition of petascale computational resources, and the evolving customer usage model. Consideration of these challenges will play a large role in solving the four technical challenges outlined below, which are presented in priority order.



Combined ASC compute environment.

2.1.1 CONCERN: Slow Development of High-Speed Type I IP Encryptors

Type I encryptor development has traditionally lagged the needs of the HPC community. As the pace of HPC development quickens, this lag continues to grow. The bulk of deployed Type I IP encryptors are still at 100 Mbps with 1Gbps representing only a fraction of the units deployed. Ten-Gbps Type I encryptors were not available for delivery until September of FY07. The lag between 10-gigabit Ethernet availability versus 10-gigabit Ethernet encryptor availability was about five years. One-Gbps encryptors are currently deployed in a parallel configuration to increase the available bandwidth, but this configuration significantly lowers overall reliability. One-Gbps encryptors do not satisfy the needs of the ASC HPC community and will be supplanted by 10-Gbps encryptors as funding permits. The needs of the HPC community are diverging from the broader market, and the encryptor vendors are not inclined to develop products focused on the small Tri-lab community. The slow development of IP-based Type I encryptors is having a negative impact on the cost and efficiency of the current ASC WAN, and the impact will only worsen for the petascale network environment. We predict that a petascale network environment could utilize 40-Gbps encryptors within five years. This challenge will be hard to solve, with the most likely successful approach to be to partner with the larger community of the DOD and DOE to drive vendor development efforts.

2.1.1.1 STRATEGY: Faster Development of High-Speed Type I IP Encryptors

Priority: Highly Important **Difficulty: Hard** **Cost: $ to $$$**

The Tri-lab community should take a more direct approach to obtaining the required encryptor technology. There are three levels of effort that the Tri-lab community can pursue to further this goal. The first is to participate in requirements development and design reviews. The second is to

encourage and support vendors entering this market by actively testing and using products that seek to improve the encryption performance. Because the ASC community has been very active in these two efforts, we have had some influence on encryption directions and products. In the past, ASC has directly contributed to early product availability by providing development funds. This third level of effort may be the only way that encryptors greater than 10 Gbps become available in the next five years because the rest of the community has not organized to push for faster capabilities. We are currently participating in the 40/100-gigabit Ethernet working group and HAIPE standards reviews. We will continue these efforts while funding is available.

2.1.2 CONCERN: Low Reliability of Current Encrypted WAN Communications System

Type I encryptors are complicated devices in their own right and represent significant operational challenges to the Tri-lab community. However, the community has had to deploy them in parallel configurations in order to deliver the performance required, which only exacerbates the operational challenges and reliability of the network. Installing multiple parallel data paths is the solution adopted by the ASC WAN community to obtain higher bandwidth than is achievable with the current 1-Gbps IP encryptor. Unfortunately, this configuration decreases the reliability of the services because a failure in any of the parallel units causes total link failure. At the current time, encryptor failure is one of the highest failure probabilities for the ASC WAN. There is a need to develop and deploy a network scheme that can tolerate one or more encryptor failures while continuing to operate at a lower level of throughput. Adapting existing networking protocols appears to be a viable strategy to achieving this goal because industry is unlikely to provide the needed solution.

2.1.2.1 STRATEGY: Increase Reliability of Encrypted WAN Network Communications System

Priority: Essential **Difficulty: Medium** **Cost: $**

To address the reliability issue involves taking a system-wide view of the entire ASC WAN environment and looking for a set of technologies that can be combined to provide significantly increased reliability. One of the proposed elements of a system solution is to apply existing network protocols in ways to ameliorate the problem of encryptor failure. In addition, limited network hardware redundancy can be installed to eliminate certain single points of failure. The developed system solutions could be immediately applied to the existing ASC WAN, thus providing benefits to the entire HPC community. Indeed, to realize the full potential of these solutions, the deployment of the developed solution would have to be Tri-lab wide.

2.1.3 CONCERN: Maintaining Efficiency and Performance While Scaling Network Protocols

Encryptors are not the only technologies that have to be stretched to meet the needs of the Tri-lab community. The TCP/IP protocol suite that is the basis of contemporary computer networking was not designed for the WAN envisioned for the petascale environment. Again, parallelism is used to provide the performance required, and operational difficulties are increased and reliable performance is decreased. Recent network protocol developments appear to provide a sound basis for increasing performance in the petascale WAN environment. However, standard industry development is unlikely to provide the required capability because the anticipated petascale WAN environment is inconsistent with standard industry deployments.

2.1.3.1 STRATEGY: Scale Network Protocols to Maintaining Efficiency for Petascale

Priority: Essential **Difficulty: Medium** **Cost: $$**

The basic data networking protocols underlying the ASC WAN were developed over 20 years ago for a vastly different environment than that of the ASC WAN. The operating characteristics of these protocols in the WAN environment limit performance; however, recent developments provide opportunities to improve the situation. First, a number of additional performance tuning algorithms have been added to the protocol implementations. These new algorithms have the potential to improve performance if they can be tuned for the ASC WAN environment. Second, the protocol suite has been increased with new standard protocols deployed. These protocols have the potential to provide increased reliability in the parallelized ASC WAN. A Tri-lab development effort to test these alternative technologies would yield immediate benefits to the ASC WAN and has the potential to deliver the performance needed for the coming petascale network environment.

Current Efforts: Modeling Collaboration with Kansas State University

An ongoing, multiyear collaboration with Kansas State University has produced a mathematical model of the ASC WAN that has been validated with laboratory tests and ACS WAN performance figures. This model incorporates the unique characteristics of the ASC WAN and is accurate enough to predict WAN performance in response to network trouble and varying traffic loads. A model of an alternative transport protocol, the Stream Control Transport Protocol (SCTP), has been developed to ascertain its performance in the ASC environment. Additional work is proposed to examine other alternative network protocols and technologies.

2.1.4 CONCERN: Managing, Operating, and Monitoring New Bulk Bandwidth Technologies

Given the nature of the budget challenges and increasing performance demands of the petascale computing environment, new or novel solutions for delivering the bandwidth required must be examined. In the past the Tri-lab community has benefited from the dedicated network bandwidth available. However, it would only be prudent to examine alternatives such as bandwidth-on-demand or shared bandwidth offerings as ways to deliver the bandwidth required by the petascale computing environment. Bandwidth-on-demand has not been offered to the Tri-lab community in response to any of the service requests sent to the commercial service providers. It will take a special effort, working with the providers, to explore this service as an option. Similarly, it will take a special effort to define the needs of the Tri-lab ASC WAN community in terms that could be addressed by a shared network environment because QoS parameters would have to be defined.

2.1.4.1 STRATEGY: Develop Techniques for New Bulk Bandwidth Technologies

Priority: Important **Difficulty: Medium** **Cost: $$**

The cost of the existing network is driven by the requirement to field very high bandwidth network links at all times so that the bandwidth is available when needed. This capability has been achieved through the installation of dedicated, high-speed communications links. Given the probable budget pressures, it is prudent to examine alternate bandwidth solutions such as bandwidth-on-demand or shared facilities. Such solutions will involve working with vendors and national test beds to develop these services and the associated management capabilities to improve reliability and performance. Shared bandwidth services would require the

implementation of end-to-end QoS if the ASC WAN was to maintain its current high performance. Deploying end-to-end QoS will require close collaborative development between the Tri-lab community and potential bandwidth providers. In a similar vein, it will take collaboration between the Tri-lab community and the vendors to develop a bandwidth-on-demand service model. If these or other new services can be deployed in the ASC WAN they could be used immediately and provide significant cost advantages.

2.2 RESOURCE INTERCONNECT

The resource interconnect consists of the very large network that connects compute platforms to parallel file systems, visualization platforms, pre- and post-processing machines, archival systems, other compute platforms, and the WAN interconnect to the other sites. Each laboratory has unique implementations of this network, but they share important common characteristics: there are hundreds to a few thousand ports, some form of parallel networking is required to build the scale required, and Ethernet is the existing technology because it scales in distance (fiber) and is common to all platforms. Each lab is also experiencing the same difficulties in this environment.

2.2.1 CONCERN: Switching Technologies Not Scaling to Size Required by Resource Interconnect

To meet the required bandwidths of petascale systems, the resource interconnect will need to scale to a few thousand 10-Gbps ports. In the next few years, the largest Ethernet switches from the major network companies will only be on the order of 512 ports. It is prohibitively expensive to build thousand port networks from these types of switches. A large portion of this expense is expended in the cabling of the too small building blocks into large useful switches. Using the fat-tree topology, half of the cables and two thirds of the ports provide no useful benefit except to provide the bisectional bandwidth of the needed ports. There are a few small companies that are starting to present ideas for building switch fabrics that they claim can grow to a few thousand ports. It is not clear that these technologies can scale or that the companies are stable enough to produce production quality equipment. There are other technologies besides Ethernet that are now being considered for the resource interconnect. These technologies have not been demonstrated to scale for the types of sustained flows in the resource interconnect networks. They also have significant distance limitations that must be overcome to be seriously considered for networks that scale beyond a few tens of meters.

2.2.1.1 STRATEGY: Develop Switching Technologies to Meet Resource Interconnect Requirements

Priority: Essential **Difficulty: Hard** **Cost: $$$**

Because the resource interconnect environment is heterogeneous and composed of several platforms, most likely provided by several different vendors, standards-based solutions are required. The two standards that are currently competing in this space are Ethernet and InfiniBand (IB). It is not clear at this point which technology will be most effective. Switch infrastructures with hundreds of high-speed ports (≥10 Gbps) are needed today. This requirement will grow to a few thousand ports for balanced petascale environments.

As a risk mitigation strategy and because we will undoubtedly use both technologies in other environments, we should pursue scaling both technologies for the resource interconnect environment. The Tri-lab community must collaborate with smaller Ethernet vendors who are

more willing to develop cost-effective solutions to the switch scaling problem. The large switch vendors have shown no real interest in providing devices that reduce the margin they are currently receiving for their products. The smaller vendors are starting to develop novel technologies that potentially allow for seamless scaling to thousands of ports. We can accelerate this process by validating and demonstrating the capabilities of the new technologies and using advanced products where possible. Our collaboration with Woven, Chelsio, and NetEffect is demonstrating the potential benefit of dynamic routing for scalable switching.

Although the IB technology has demonstrated large port count networks, it has also demonstrated significant performance issues at those scales. The static nature of the internal routing induces significant link oversubscription. TCP/IP over IB does not perform as well as is required. The RDMA protocol that is native to IB is extremely efficient, but many of the data movement tools currently employed in the ASC network cannot utilize RDMA. The Tri-labs have been one of the prime motivators and funding sources of the IB standardization process. There are sufficient benefits to IB that warrant the continued effort of the Tri-labs to develop the standard, develop software tools that can use IB, and investigate system designs that can take full advantage of the IB technologies.

2.2.2 CONCERN: Inefficiency of Network Interface Cards and Protocols

Although host performance is continuously improving, there is always a period of time where hosts are unable to completely utilize the latest high-speed NIC (network interface card) with TCP/IP data flows. We are currently at the point where the latest hosts are fairly well matched to a single 10-gigabit Ethernet interface when using standard TCP/IP. The most common host configuration in the resource interconnect environment is that of a gateway node between the Internal Interconnect of a platform and the resource interconnect network. In these gateway nodes, the host is moving data on the internal interconnect as well as the resource interconnect, thus doubling the amount of data the host must move. Sustained throughput drops significantly in this scenario. The availability of sufficient bus bandwidth also limits the sustained throughput capabilities of these gateway nodes. Without sufficient hardware bandwidth, the protocols cannot possibly achieve our required performance. There are several mechanisms that have been proposed to raise the efficiency of data movement through hosts. Because most of these mechanisms impact the data-movement applications, care must be taken to ensure the system as a whole continues to provide all of the necessary functionality as more efficient mechanisms are developed and deployed.

2.2.2.1 STRATEGY: Develop Technologies that Maximize Effective Bandwidth of NICs

Priority: Highly Important **Difficulty: Hard** **Cost: $$**

There are two main paths to pursue to improve the efficiency of data throughput in our hosts. Becasue TCP/IP is the dominant protocol for the current resource interconnect and Ethernet technologies, we need to work with the Linux community to improve the efficiency of TCP/IP processing. There are a few different mechanisms that appear to have potential for improvements including splice, large segment offload, user space TCP, and TCP offload engines. The Tri-lab community should continue to participate with the industry to complete the work on these mechanisms and ensure that they will function in our environments.

The other potential path to very high efficiencies is to modify our environment to take full advantage of the RDMA protocol. The existing RDMA standard is an extremely efficient mode

of data movement, but it does not work the same way as TCP/IP. This requires extensive modification of many of our existing data movement applications. It is not clear that these modifications would be possible due to both technical and vendor proprietary issues. There are other data movement applications that are already optimized for the RDMA protocol. The Tri-lab community must investigate our ability to utilize RDMA and provide all of the required functionality. This is even more important now that the RDMA protocol can be utilized with Ethernet technologies as well as was demonstrated by our Woven, Chelsio, and NetEffect collaboration. The Tri-lab community must also work with the motherboard vendors to ensure that there is sufficient bandwidth in the I/O busses. There are new bus technologies such as PCIe v2 and HT 3 forthcoming in the industry. Those or other faster technologies must be deployed for the petascale environment.

2.2.3 CONCERN: Managing Resource Interconnect Networks and Potential New Technologies

As the size of the resource interconnect networks grow, the difficulties of management and operations will grow as well. Given the extremely large count of switch ports, physical infrastructure, NICs, and hosts that will be connected to the network, the probability that there will be failures at any given time is extremely high. This is the same problem experienced in the compute platforms with failures in the nodes, interconnects, and disks. The resource interconnect environment must be designed for resiliency in the presence of failures anywhere in the system. Automated tools for detecting, isolating, and notification of failures will be critical to providing a reasonable mean time to repair for the system. If new technologies and protocols are deployed, then new mechanisms, techniques, and even testing products will be required.

2.2.3.1 STRATEGY: Develop Techniques for Resource Interconnect Management Technologies

Priority: Important **Difficulty: Medium** **Cost: $$**

The combined numbers of devices that will exist in the petascale resource interconnect will be a serious challenge for any operational tools and processes. Automated tools for detecting and diagnosing failures must be developed and deployed in such a large environment. There will definitely be new technologies in this environment that will require new tool development. This could even include hardware devices for IB links, dynamic routing monitoring and debugging tools, etc. We will partner with vendors to ensure products are available that meet our requirements.

Management of the resource interconnect network will require much more than just monitoring the health of individual links and components. The complexity of the parallel architectures that will be implemented on top of the physical network will be significant. The Tri-lab community must develop tools that will closely monitor the combined parallel communications to ensure that the system is performing as designed. The goal of system resiliency to failures in the resource interconnect is not solely a function of network components. Architectures and applications that continue to provide critical services, even when there are failed components, require system-wide integration of applications and hardware. The networking team must work with the application developers and vendors to develop fault-tolerance at the network level. This is a primary focus of the NITRD research agenda, and ASC should participate and utilize their efforts wherever possible.

2.3 INTERNAL INTERCONNECT

For all but the most embarrassingly parallel applications, the ability of the internal high-speed interconnect must be balanced with the rest of the platform architecture to ensure good performance at scale. The transition from terascale platforms to petascale platforms is being achieved through higher parallelism within a compute node and by increasing the total number of nodes within a system. Both trends drive the need to increase the performance of traditional high-speed interconnect metrics while creating new requirements that were not significant in the terascale era.

2.3.1 CONCERN: Topologies and Node-to-Node Scaling

As HPC systems increase in size, the complexity of each node increases as does the complexity of the chips within the node. The simple conceptual model of CPU and NIC does not reflect the new reality of a multi-layer network, which includes "network on chip," bridge chips, and the NIC gateway to the "internal interconnect." We need to influence and direct all three of these technology areas toward useful HPC solutions if we are to satisfy the requirements of petascale internal interconnects.

Two internal interconnect topologies are most commonly found in the ASC complex, some variant of a fat-tree and a 3D mesh (or torus). Although the 3D mesh is capable of scaling as the size of the system grows, it is not a topology supported by the commodity interconnect vendors. This is primarily due to market drivers. Most HPC platforms in the global market are small enough that the fat-tree topology is preferred due to its better inherent characteristics of higher bisection bandwidth and lower diameter than a mesh. However, as the size (number of endpoints) of the system grows, the number of switch components and cables grows logarithmically, thus causing the system cost to increase proportionately. In addition to cost, the complexity of the physical layout and the distance between nodes of the machine becomes a limiting factor.

It is essential that the Tri-labs understand the impact such topologies will have on application performance, necessitating the need for robust modeling and performance analysis tools.

2.3.1.1 STRATEGY: Development of Performance Modeling and Analysis Tools to Judge the Impact of Changes to Key Architectural Features

Priority: Highly Important **Difficulty: Medium** **Cost: $$$ to $**

Although the Tri-labs, industry, and academia have some modeling and analysis capabilities, in many ways it is insufficient to address detailed network analysis, especially at large scales. The Tri-labs need to increase their level of investment in this area in order to better define requirements, metrics, and future acquisitions.

2.3.2 CONCERN: Decreasing Byte-to-FLOPs Ratios

Increasing the level of parallelism on a node is the new catalyst that will ensure Moore's law will continue to be realized in the coming years. As the FLOPs per node increases at a rate greater than the performance of the interconnect, the computation-to-communication ratio decreases and hence the efficiency of the platform drops.

A desirable byte-to-FLOPs (B/F) ratio is 1.0. Initial petascale platforms will have a B/F of ~0.06, and some will be an order of magnitude more or less. Experience has shown that these machines will have difficulty scaling to more than a few thousand nodes on ASC workloads. This will be acceptable for capacity workloads, but capability workloads will not be possible. Bandwidths can be increased through several strategies: increasing the number of pins per port, increasing the bandwidth per port, and having multiple NICs per node (which increases the topological complexity and introduces new problems such as effective striping techniques, out of order delivery, etc.).

2.3.2.1 STRATEGY: Place Higher Emphasis on Internal Network Requirements in Future Petascale Acquisitions

Priority: Highly Important **Difficulty: Medium** **Cost: $$$ to $**

Although the Tri-labs have been successful in deploying platforms with very good scaling characteristics, it will become imperative to push the vendors even harder in future acquisitions. As has been stated many times above, the B/F ratios are decreasing rapidly due to advancing processor developments, and the external market is not requiring high-speed interconnects at the level required for Tri-lab workloads. The Tri-labs can do this by developing a detailed set of requirements for future machines and providing realistic application and micro-benchmarks to measure performance metrics in as close to a real-world scenario as possible.

2.3.3 CONCERN: Latency and Message Throughput

Latency of small messages is a factor limiting many of our applications, and this trait is likely to become even more dominant when applications can no longer scale in size (i.e., memory) as the size of the job increases. This limitation is likely to occur in the next several years; therefore, it is strategically important that the HPC community understand all of the latency issues in order to optimize designs to reduce latency. This includes application library layers such as MPI and newer programming paradigms that show potential for increasing use within ASC, such as UPC.

Messaging rate is a metric that has been given increasing attention recently. A few things are driving this, but the most prevalent reason is the introduction of multicore chips and hence multiple outstanding messaging commands being simultaneously served by a single NIC. For the most part, current high-speed interconnects (commodity or custom) have not been designed for high messaging rates.

2.3.3.1 STRATEGY: Develop Technologies and Metrics for Reduced Latency and Improved Message Injection Rate

Priority: Essential **Difficulty: High** **Cost: $$$**

There are significant projected improvements in the bandwidth and latency of the next generation of computing cores. As these improvements become available, the Tri-lab community needs to help develop the supporting codes, algorithms, and interface cards to ensure that these improvements benefit the ASC code performance. Further socialization with vendors on the importance of increasing the messaging rate is required. Standardization of a meaningful messaging rate benchmark should be established across the Tri-labs in order to provide meaningful comparisons.

2.3.4 CONCERN: Scaling Limitations of Industry Standardized InfiniBand Solutions

Current IB silicon only implements reliable connections; however, in general, connection based protocols are inherently not scalable as the resources required grow linearly with the size of the machine. Another weakness of current IB implementations is the use of static routing algorithms that inevitably create “hot spots” in the network. Hot spots are congestion points that lead to poor performance.

IB has gained a significant market share in HPC, and as a result has also become a major commodity interconnect vendor actively pushing data rates and features necessary for large-scale HPC. The sweet spot of the market is, however, much smaller in scale than the needs of the Tri-labs; hence, it continues to be difficult to convince the IB community to develop technologies to address the top 1% of the market. The Tri-labs should continue to work with commodity network vendors to better address unique Tri-lab requirements.

Features not found in many interconnect products currently available that are required for tens of petaFLOPs platforms include:

- Hardware-based global communication support. This is an application requirement and given the scale “system noise” will otherwise diminish significantly the performance of global communications that dominates many applications.

- A global clock or some other method of heartbeat in the network to help synchronize processes for noise reduction.

- Communication stack offload to ensure a low communication/computation ratio and hence high processor utilization during communications. This can include new features in the NIC for global reductions, e.g., floating point processing for global reductions.

While the IB specification is an open standard and current software development efforts are open source projects, there is still only a single major IB silicon company, Mellanox.

2.3.4.1 STRATEGY: Partner with Key Industry Providers to Stimulate Development of Features and Performance beyond the Mainstream Market

Priority: Essential **Difficulty: Hard to Medium** **Cost: $$$ to $$**

The Tri-labs have historically been very successful in influencing and, in some cases, driving developments in high-speed networks (both hardware and software). Examples include the Cray XT3/4 network, IB, and Myrinet. To ensure that future petascale architectures scale, it will be essential that we continue to work with vendors to push interconnect capabilities beyond what is required for the bulk of the HPC market. Industry by itself will continue to push processor architectures, which will provide the necessary FLOPs and hence petascale machines, but high-speed interconnects to tie together tens of thousands of such processors in an efficient matter is something that the DOE must drive. As such, we should establish strategic development programs with industry partners that specifically target high-speed interconnects. The Tri-lab community must also work with the motherboard and chip vendors to ensure that there is sufficient bandwidth in the I/O busses including “network on chip” and board level bridge chips.

There are new bridge technologies such as PCIe v2 and HT 3 forthcoming in the industry. Those or other faster technologies must be deployed for the petascale environment.

Because IB will continue to be a major player in the high-performance interconnect market, we should continue to work with that community to provide better capabilities that meet our requirements. A better connection solution is to have a protocol that allocates resources based on demand, with a limit being placed on the total size. Mellanox and the IB community are working on methods to minimize the impact of the reliable connection. This includes shared receive queues and dynamic allocation semantics. However, these efforts are immature and need significant testing at large scale, which is difficult for any development group to do. The Tri-labs have been encouraging the IB community to investigate dynamic or adaptive routing technologies. The Tri-labs should continue to work with this community to provide guidance and access to platforms for large-scale testing. Fostering broader competition within the IB community at the silicon manufacturer level may promote a healthier industry where Tri-lab needs are more easily met due to competitive forces.

2.3.5 CONCERN: Managing Scaled Interconnect Networks and Potential New Technologies

With the scale of the network growing to tens of thousands of end-points, the need for management tools to analyze and debug network issues is a must. These tools must also support reading of registers and other performance counters in the system for performance analysis. They must also be robust and resilient to failure. As mentioned in many of the areas discussed above, performance modeling of future internal interconnect architectures will require sophisticated performance modeling tools in order to best understand the impact that design decisions will have on the Tri-lab workloads. All of the issues discussed above are applicable to each of the Tri-labs and in general will provide for national leadership in supercomputing, which will benefit the entire community. ASC has a strong history and track record in providing national leadership and must continue to do so. Although the issues are common, each laboratory's requirements may not be of the same magnitude for any given capability. Each lab must continue to work with particular vendors to fit their requirement in addition to those of the other labs.

2.3.5.1 STRATEGY: Development of Management and Performance Analysis Tools for Compute Interconnect

Priority: Important **Difficulty: Medium** **Cost: $$**

The Tri-labs need to work with industry to develop a tool or a set of tools to aid management and performance analysis at the network level. Current tools are non-existent or insufficient. This will also require working with vendors at the silicon level to define performance counters that need to be integrated into next generation NICs and switches.

3 TIMELINE SUMMARY

Strategy	Activity	Near Term	Medium Term	Long Term	Comments
2.1.1.1	Faster development of high-speed Type I IP encryptors				
2.1.2.1	Increase reliability of encrypted WAN network communications system				
2.1.3.1	Scale Network Protocols to Maintaining Efficiency for Petascale				
2.1.4.1	Develop techniques for new bulk bandwidth technologies				Phase 2: Tools to monitor and respond to hardware errors
2.2.1.1	Develop switching technologies to meet resource interconnect requirements				
2.2.2.1	Develop technologies that maximize effective bandwidth of NICs				
2.2.3.1	Develop techniques for resource interconnect management technologies				
2.3.1.1	Development of performance modeling and analysis tools to judge the impact of changes to key architectural features				
2.3.2.1	Place higher emphasis on internal network requirements in future petascale acquisitions				
2.3.3.1	Develop technologies and metrics for reduced latency and improved message injection rate				
2.3.4.1	Partner with key industry providers to stimulate development of features and performance beyond the mainstream market				
2.3.5.1	Development of management and performance analysis tools for compute interconnect				

SUMMARY AND CONCLUSIONS

1 OVERALL INFRASTRUCTURE STRATEGY

This document presents clarification and answers to questions raised in an initial February 2007 meeting to discuss and plan the infrastructures needed for coming ASC petascale environments.

- What are the most important components that must exist for a successful petascale environment, and do any of them overlap technical CSSE/FOUS areas?
- Are there barriers, gaps, or issues that must be addressed to develop or deploy these components, and what are the user concerns that motivate these concerns?
- What are the approaches that can be taken to address those barriers, gaps, or issues for petascale, and might these approaches be usable and/or relevant outside ASC?

Supercomputing infrastructure can be viewed as an ecosystem. Organisms in such an ecosystem are the technologies and components that mutually reinforce each other and the overall stability of the ecosystem. They must adapt and evolve to maintain the health of the ecosystem. Using this analogy, CSSE/FOUS can be viewed as a major piece (or several pieces) of an overall ASC computational ecosystem, together with users, codes, algorithms, and other key "organisms." Surviving and flourishing within such a system requires careful observations, continuous monitoring, and informed decisions regarding choices of technology to maintain an appropriate balance.

Computers today are much faster but harder to use. We have moved from sequential codes and vector capability to large parallel clusters, both homogeneous and hybrid. The number of cores continues to increase; however, processors are only a part of an ecosystem. A balanced petascale ecosystem has requirements that are not necessarily part of lower-end commercial systems and must be driven by (or adapted by) ASC capability and advanced architectures. As described in this document, example areas (or organisms) required by an ASC petascale ecosystem are programming environments and tools, petascale data analysis, I/O file systems and storage, and networks and interconnects.

Without federal investment as a forcing function, the computing industry will not evolve to usable petascale computing systems in the time required for effective and responsible nuclear weapons stewardship. Hence, NNSA (and others) will continue to be a major driver for high-end technology. ASC and NNSA recognize that the nuclear weapons budget in the future is expected to remain constant (or even decline). NNSA's Complex Transformation vision states that level funding is an overarching constraint for planning purposes, and that should budgets decline, some programs will need to be protected at the expense of schedule, scope, or increased risk. An implication for ASC is that the Tri-labs are likely to be encouraged to invest in new hardware and software architectural directions in partnership with other federal agencies and computer vendors, whose business plans these investments can leverage, and to develop strategies for productivity gains within realistic future budgets.

2 CROSSCUTTING TECHNICAL CONCERNS

There are likely several technical concerns that crosscut the four technical working groups organized for creating this document. Three specific concerns were identified by the working groups and the CSSE/FOUS managers.

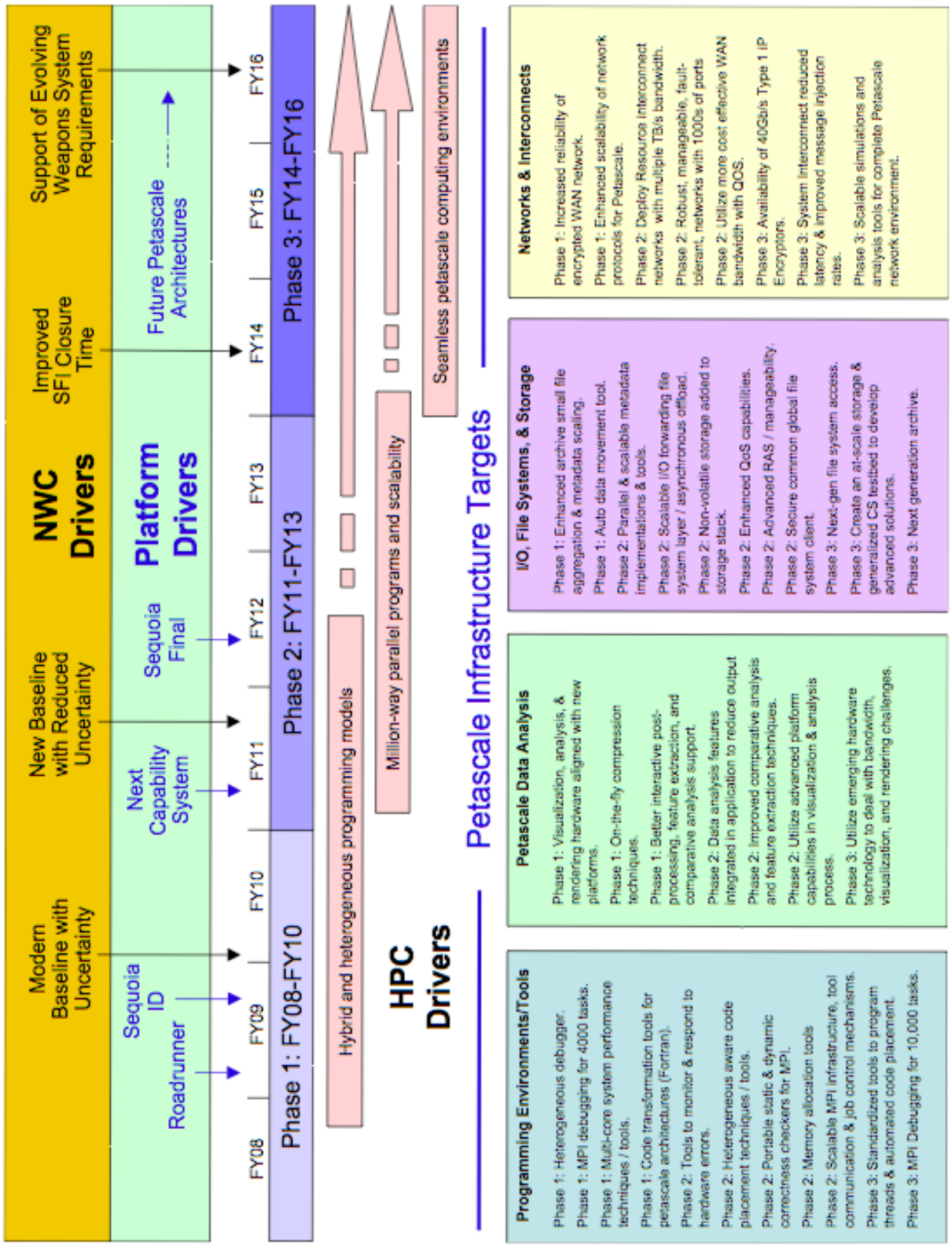
- Infrastructure component management and system administration concerns.
- Linkages between programming model choices and internal interconnects.
- Linkages between data analysis capabilities and I/O bandwidth constraints.

The process of infrastructure component failure detection, identification, and fix is time consuming. Future infrastructure systems will need to monitor themselves, automatically identify typical failures, and initiate corrective action. This kind of “autonomic” behavior will be essential to operate petascale environments. Work is starting on steps to integrate data feeds into common monitoring frameworks to speed problem detection and identification by making relevant data sets available to key personnel and to develop tools to track application level test results over time. In the future, monitoring must be enhanced with scripts capable of taking corrective action when frequently encountered problems are detected.

There are linkages between programming models and internal interconnection networks. The strategy to investigate and develop new programming models will have an associated cost that may drive investments made in the internal interconnect area to extend the life of ASC MPI application codes. A return on investment analysis should be performed to understand the tradeoffs and potential impact of the effort that could/should be expended in these areas. But the real cost tradeoff extends beyond the scope of CSSE/FOUS. Because of the cost of rewriting our code base, there may be motivation for a significant investment in internal interconnect if this investment allows us to extend the useful life of our MPI code base.

Constraints on I/O will affect the amount of data that can be stored on secondary and tertiary storage for a given simulation (computation speed will far outpace our ability to write out data) and the speed with which the data can be visualized. Reading data from disk will constrain petascale data analysis capabilities—everything from switching between time steps in a visualization tool to comparing the results of a single time-step in two simulations. Petascale infrastructures will demand that integrated solutions be designed and delivered in concert so that petascale results can be analyzed. Note that this includes not only the platform-specific visualization and analysis resource and the rendering resource, but also the storage, I/O system, and the LANs and WANs connecting the compute platforms to the desktop or the visualization cluster. These will all need to be scaled to match the increase in the amount of data.

3 FY2008–FY2016 PLANNING TIMELINE



Overview of petascale infrastructure drivers and targets.

4 LONG-TERM VIEW AND NEXT STEPS

This plan is a formal deliverable for an ASC CSSE/FOUS Level 2 Milestone. It acknowledges and quantifies technical gaps or issues for petascale infrastructure, and, where such gaps exist, defines approaches to closing them. While the formal milestone deliverable (this document) will be completed in March 2008, petascale infrastructure components that it describes will be developed and deployed throughout a decade-long time frame. The plan is applicable to multiple ASC petascale platforms deployed during that time, including Roadrunner, Sequoia ID and final systems, and other potential petascale platforms as further described in the recent *ASC Platform Strategy* document.

This Plan will be used as technical guidance to CSSE/FOUS and senior ASC program managers to better inform detailed program planning. Additionally, it will be used to coordinate goals and objectives in separate parts of the ASC program. CSSE/FOUS managers and the technical working groups will annually review the plan and update the targets. The plan will be more extensively updated at the beginning of Phase II (early 2011) and at the beginning of Phase III (early 2014) to reflect ongoing technical progress, ASC platform architectural directions, and newly uncovered computational issues. While updated plans may not necessarily be Level 2 Milestones, this regular review and update process is deemed to be necessary to ensure that infrastructure planning stays relevant.

Not all strategies and approaches described in this document will necessarily pertain to all platforms. Because of differences in architecture, some technical approaches may be more relevant to hybrid or heterogeneous petascale platforms (e.g., Roadrunner). Others may be more relevant to more homogeneous multi-core architectures such as Sequoia. Furthermore, as shown in the *ASC Roadmap* and in the above planning timeline, a “seamless” petascale computing environment is an ASC target goal for the coming decade. CSSE/FOUS program executives will need to ensure the infrastructures to be developed and deployed are consistent with this seamless environment approach, NWC programmatic drivers, ASC petascale platforms and application requirements.

APPENDICES

A. ACKNOWLEDGMENTS

Lab Points of Contact

Steve Louis, LLNL

John Naegle, SNL

Bob Tomlinson, LANL

DOE NNSA and ASCR

Thuc Hoang, NNSA

Fred Johnson, ASCR

Sander Lee, NNSA

Tina Macaluso, NNSA

Bob Meisner, NNSA

ASC Users

Tom Brunner, SNL

John Daly, LANL

Brian Pudliner, LLNL

Jim Stewart, SNL

Bob Webster, LANL

Mike Zika, LLNL

Programming Environments and Tools

Bronis de Supinski (Lead), LLNL

Jim Ang, SNL

Scott Futral, LLNL

Curtis Janssen, SNL

Ken Koch, LANL

Dave Montoya, LANL

Craig Rasmussen, LANL

Karl-Heinz Winkler, LANL

Petascale Data Analysis

David Rogers (Lead), SNL

Jim Ahrens, LANL

Eric Brugger, LLNL

Jerry Friesen, SNL

Bob Kares, LANL

Laura Monroe, LANL

Dino Pavlakos, SNL

John Thorp, LANL

I/O, File Systems, Storage

Gary Grider (Lead), LANL

Mark Gary, LLNL

Bill Loewe, LLNL

Susan McRee, SNL

James Nunez, LANL

Jerry Shoopman, LLNL

Judy Sturtevant, SNL

Lee Ward, SNL

Networks and Interconnects

John Naegle (Lead), SNL

Parks Fields, LANL

Adolfy Hoisie, LANL

Richard Hu, SNL

Bryan Lawver, LLNL

Tim Merrigan, LANL

Denny Rice, LANL

Larry Tolendino, SNL

Dave Wiltzius, LLNL

B. ACRONYMS

API	Application Programming Interface
ASC	Advanced Simulation and Computing
ASCI	Accelerated Strategic Computing Initiative
CHAOS	Clustered High Availability Operating System
CMOS	Complementary-[Symmetry] Metal–Oxide–Semiconductor
CSSE	Computational Systems and Software Environment
DP	Defense Programs
DVI	Digital Visual Interface
FOUS	Facility Operations and User Support
HDF	Hierarchical Data Format
HPSS	High Performance Storage System
IP	Internet Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MPI	Message Passing Interface
NIC	Network Interface Card
NNSA	National Nuclear Security Administration
NUMA	Non-Uniform Memory Access/Architecture
PnetCDF	Parallel network Common Data Format
PVFS	Parallel Virtual File System
QMU	Quantification of Margins and Uncertainties
QoS	Quality of Service

RAID	Redundant Arrays of Independent Disks
RAS	Reliability, Availability, and Serviceability
RDMA	Remote Direct Memory Access
SIMD	Single Instruction, Multiple Data
SSP	Stockpile Stewardship Program
V&V	Verification and Validation
WAN	Wide Area Network
WBS	Work Breakdown Structure

C. REFERENCES

1. ASC Platform Strategy.
   http://sandia.gov/NNSA/ASC/pdfs/AscPlatform2007.pdf

2. ASC Strategy: The Next Ten Years.
   http://sandia.gov/NNSA/ASC/pdfs/Strat10yr_MT.pdf

3. ASC Business Model.
   http://sandia.gov/NNSA/ASC/pdfs/ASC-Bus-Mod-2005-w.pdf

4. ASCI Technology Prospectus: Simulation and Computational Science.
   http://sandia.gov/NNSA/ASC/pdfs/prospectus.pdf

5. Getting Up to Speed: The Future of Supercomputing.
   http://www7.nationalacademies.org/cstb/pub_supercomp.html

6. NNSA Strategic Plan: DOE/NA-0010.
   http://www.sandia.gov/asc/pubs_pres/pubs/NNSA_Strategic_Plan_Nov_2004.pdf

7. JASON 2003 Report: Requirements for ASCI.
   http://www.sandia.gov/asc/pubs_pres/pubs/ASCI_Requirements_091503.pdf

8. Software Development Tools for Petascale Computing Workshop Report.
   http://www.csm.ornl.gov/workshops/Petascale07/sdtpc_workshop_report.pdf

9. Petascale Systems Integration into Large Scale Facilities Workshop Report.
   http://www.nersc.gov/projects/HPC-Integration/PetascaleFinalDraft.doc

10. High End Computing Revitalization Task Force (HECRTF), Inter Agency Working Group File Systems and I/O Research Workshop HECIWG FSIO 2006.
    http://institute.lanl.gov/hec-fsio/docs/HECIWG-FSIO-FY06-Workshop-Document-FINAL-FINAL.pdf

11. Roadrunner ASC Platform Statement of Work (Contract Number 34851-001-06, Appendix A, Version 3), Internal Document.

12. Los Alamos National Laboratory Advanced Simulation and Computing (ASC) Five-Year High Performance Computing (HPC) Plan, 2007, Internal Document.

13. LLNL FY07 I/O Integration Blueprint (UCRL-TR-228502), Internal Document.

14. The Grand Challenge of Managing the Petascale Facility, Argonne ANL/MCS-07/5
    ftp://info.mcs.anl.gov/pub/tech_reports/reports/ANL-MCS-07-5.pdf

D. L2 MILESTONE TEXT

Milestone (ID#): Infrastructure deployment plan for ASC petascale environments
Level: 2
Fiscal Year: FY08
DOE Area/Campaign: ASC
Completion Date: Mar-08
ASC nWBS Subprogram: Computational Systems and Software Environment (CSSE) & Facility Operations and User Support (FOUS)
Participating Sites: LLNL, LANL, SNL
Participating Programs/Campaigns: ASC
Description: The ASC Petascale Environment Infrastructure Deployment Plan will identify, assess, and specify the development and deployment approaches for critical components in four different technical areas: (1) development environment and tools; (2) petascale data analysis; (3) I/O, file systems and archives; and (4) networks and interconnects. This Plan will identify and quantify potential technical gaps or issues, and, where they exist, will define a prioritized approach to closing those gaps. While the specific deliverable (a planning document) for this milestone is to be completed in Q2 FY08, the petascale infrastructure components will likely be deployed throughout a five-year FY08–FY12 timetable, and prioritization given to enabling predictive weapons simulations. The plan will be applicable to multiple ASC petascale platforms deployed during that time, including Roadrunner and the Sequoia ID and final systems.
Completion Criteria: This milestone will produce an integrated Tri-lab planning document that can be applied to multiple ASC petascale environments over multiple years. While initial petascale infrastructures and capabilities may be site or platform specific, the eventual goal as supported by this milestone is to provide a seamless petascale user environment for capability computing as envisioned and motivated by the ASC Roadmap. The deployment plan provided by this milestone will operate as a stand-alone document, but it may be desirable to also incorporate, by reference or as appendices, additional Tri-lab or lab-specific documents, for example the "ASC I/O and Storage FY06-FY10 Technology Update and Plan for HPC File Systems, Scalable I/O, and Archival Storage," and the "LLNL FY08 I/O Integration Blueprint."
Customers: Tri-lab and individual laboratory system integration teams, future users of ASC petascale computing platforms, NNSA/ASC Headquarters and ASC Execs.
Milestone Certification Method: Completion evidence for this milestone will be: (1) professional documentation - the formal plan, reviewed and released for wide (or potentially unlimited) distribution; and (2) an appropriate internal program review with documented results, incorporating any supporting presentation slides.

Supporting Resources: Tri-lab CSSE and FOUS products and project team personnel.				
Codes/Simulation Tools Employed: TBD				
Contribution to the ASC Program: Provides the detailed deployment planning for petascale computational environments and infrastructure, targeted at improving the usability and cost performance of petascale capability platforms and codes.				
Contribution to Stockpile Stewardship: Ensures successful deployment of petascale computational environments in support of overall SSP goals, including Uncertainty Quantification (UQ) analyses, advanced weapons science studies, and enhanced integrated design code predictive capabilities.				
No.	**Risk Description**	**Risk Assessment (low, medium, high)**		
		Consequence	**Likelihood**	**Exposure**
1.	Technical working groups may not reach consensus or meet planned schedules for first and final drafts of document. Risk control through Lab POCs and HQ coordination.	Low	Moderate	Low
2.	Some technical gaps may require complex multi-lab and/or multi-agency collaborative approaches, although this will likely not delay creating the planning document. Risk control through Lab POCs and HQ coordination.	Low	Moderate	Low
3.	Inability to predict the future state of some petascale technologies may cause document to not be as effective as envisioned. Technical working groups, POCs and HQ to accept this risk.	Low	Moderate	Low